

---

**МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИРОВАНИЕ  
В ПРИБОРОСТРОЕНИИ**

---

УДК 004.932;004.896

© Н. А. Водичев, Д. А. Гаврилов, А. В. Леус, В. А. Ефремов, И. В. Холодняк, В. А. Зуев, М. М. Паршиков, А. Ю. Троегубов, Н. В. Герштейн, В. Е. Лаухин, А. Е. Антипов, 2025

**АЛГОРИТМ РАСЧЕТА ПОЛОЖЕНИЯ И СКОРОСТИ ДВИЖЕНИЯ  
ОБЪЕКТОВ НА ПЛОСКОСТИ ПО ДАННЫМ С МОНОКУЛЯРНОЙ  
КАМЕРЫ**

В статье представлен алгоритм автоматического расчета матрицы гомографии, предназначенный для оценки скорости движения погрузчиков в складских условиях. Предложенный алгоритм использует видеопоток с камер наблюдения для преобразования координат объектов из перспективного изображения в метрическое пространство, что позволяет определять положения погрузчиков в реальных единицах измерения. В алгоритме реализована автоматическая калибровка на основе распознавания поддонов с известными размерами и использования характеристик камеры. Предложен алгоритм автоматического подбора параметров монокулярной системы для случая, если они заранее неизвестны. Проведен эксперимент с использованием реальных видеозаписей, демонстрирующий применимость предложенного алгоритма для мониторинга логистических процессов и повышения эффективности управления складским транспортом. Эксперимент показал, что при расчете средней скорости движения погрузчиков 98% полученных значений попадают в погрешность  $\pm 1$  км/ч.

*Ключевые слова:* вычисление скорости, компьютерное зрение, нейронные сети, матрица гомографии

**ВВЕДЕНИЕ**

Технологии компьютерного зрения активно применяются в различных областях — от автономных транспортных средств до систем безопасности. Такие системы используются для мониторинга пешеходных зон, дорог общего пользования, складских помещений и в управлении беспилотным транспортом. Данные технологии в основном применяются в области анализа окружающей среды для принятия решений в реальном времени. В частности, такие системы могут оценивать траектории движения объектов, определять их скорость и положение, а также прогнозировать потенциальные столкновения или нарушения правил [1–4]. Эти функции особенно важны в условиях высокой динамичности и ограниченного пространства, например, на складских площадках, где необходимо обеспечивать безопасное взаимодействие между погрузчиками, работниками и другими элементами инфраструктуры.

Зачастую для решения задач безопасности перемещения используется оборудование, установленное на погрузчик. В частности, задача предотвращения столкновений решается с помощью парктроников, лидаров или радаров [5, 6]. Однако использование данных технологий имеет ряд недостатков и ограничений:

- необходимость наличия на каждом погрузчике и, как следствие, высокая стоимость приобретения и установки;
- выполнение ограниченного списка задач — измерение расстояний до объектов и предотвращение столкновений;
- сложности при масштабировании.

В то же время на большинстве современных складов уже установлены системы видеонаблюдения, которые могут быть эффективно использованы для решения задач безопасности. Внедрение алгоритмов компьютерного зрения открывает новые возможности для мониторинга и предотвращения столкновений. Однако большое количество установленных камер и отличие складских помещений друг от друга приводит к значительному увеличению объемов и разнообразию данных, которые необходимо обрабатывать. В таких условиях актуальными становятся автоматизированные методы анализа, способные эффективно обрабатывать большие массивы информации и быть устойчивыми. В качестве подобного метода можно использовать искусственные нейронные сети, доказавшие свою эффективность при решении широкого круга задач [7]. Обобщающая способность нейронных сетей и простота установки монокулярных камер позволяет упростить внедрение разработанного решения в существующие склады. В сочетании с имеющимися в погрузчике технологиями

предложенный алгоритм позволит расширить список выполняемых задач. Более того, использование данной системы позволяет снизить затраты, исключая необходимость значительных инвестиций в новое оборудование.

В данной работе предлагается алгоритм вычисления положений и скоростей движения объектов на плоскости на основе данных с монокулярной камеры. Для обнаружения объектов интереса используются сверточные нейронные сети. Основной особенностью предложенного алгоритма является автоматический расчет матрицы гомографии, что позволяет обходиться без предварительной ручной калибровки камеры и использования внешней информации о наблюдаемой плоскости. Составление матрицы гомографии осуществляется с помощью распознавания объектов с известными размерами, связанными с наблюдаемой плоскостью. Основными преимуществами алгоритма являются:

- использование монокулярной системы вместо радаров и стереокамер;
- способность обрабатывать большие объемы видеоданных и потоков с различных камер;
- автоматический расчет матрицы гомографии для определения положений объектов на плоскости;
- 98% рассчитанных средних скоростей лежат в интервале 1 км/ч от эталонных значений.

## ОБЗОР ЛИТЕРАТУРЫ

Было разработано множество методов для определения положения и скоростей объектов по данным с монокулярной камеры. В работах [8–10] представлен метод расчета скоростей движения транспортных средств с использованием трех точек схода. Для определения метрических характеристик полученной сетки необходимо вычислить коэффициент масштаба. В работе [9] для этого используется статистика о средних габаритах автомобилей, которая сопоставляется с габаритами в пикселях. Это может вызывать ошибки, связанные с разнообразием данных в статистике и положением камеры. В статье [10] вместо использования данных статистики авторы предлагают классифицировать марку, модель и год выпуска автомобиля и учитывать его реальные габариты. Данный способ позволяет повысить точность вычисления коэффициента масштаба, однако сложен в реализации, т.к. для обучения классификатора требуется большой объем данных.

В научной литературе также представлены методы, которые для расчета местоположения и пройденного расстояния применяют пред-

варительную калибровку камеры на основе метаинформации о наблюдаемой части дороги [11, 12]. Недостатком такого подхода является требование доступности точных данных о наблюдаемой плоскости, что не всегда возможно.

Помимо этого, ученые предлагают алгоритмы, которые позволяют определять удаленность объектов от камеры на основе распознавания предметов с заранее известными размерами. Например, в статье [13] предлагается алгоритм, где в качестве предмета с известными размерами используется номерной знак автомобиля. Основой алгоритма выступает геометрическая модель системы, а для вычисления скорости используется изменение площади номерного знака на изображении. Основным недостатком данного подхода является необходимость распознавать номер на каждом кадре. Это может быть затруднительно, т.к. номер может быть загрязнен или не виден, а при высоких скоростях и недостаточном освещении его контуры будут размыты. В работе [14] используется сегментация транспортных средств и две зоны интереса. Для каждой зоны на основе найденной маски и истинных размеров автомобиля вычисляется коэффициент *PPM*, определяющий количество пикселей в одном метре. Далее этот коэффициент используется для расчета пройденного расстояния. Недостатком данного подхода является предварительная калибровка и вычисление коэффициента *PPM*, от которого напрямую зависит точность вычисления скорости движения объектов.

## ОПИСАНИЕ АЛГОРИТМА

### Калибровка камеры

#### Матрица гомографии

Для определения положения обнаруженного объекта на наблюдаемой плоскости необходимо построить соответствие между координатами на изображении и координатами на этой плоскости. Для этого можно воспользоваться матрицей гомографии [15]:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}. \quad (1)$$

Выражение (1) представляет собой матрицу  $3 \times 3$ , в которой компоненты  $h_{11}$ ,  $h_{12}$ ,  $h_{21}$  и  $h_{22}$  отвечают за поворот и растяжение системы координат,  $h_{13}$  и  $h_{23}$  — за сдвиг начала координат, а  $h_{31}$  и  $h_{32}$  — за перспективное искажение. Перевод вектора из одной системы координат

в другую осуществляется домножением на матрицу гомографии:

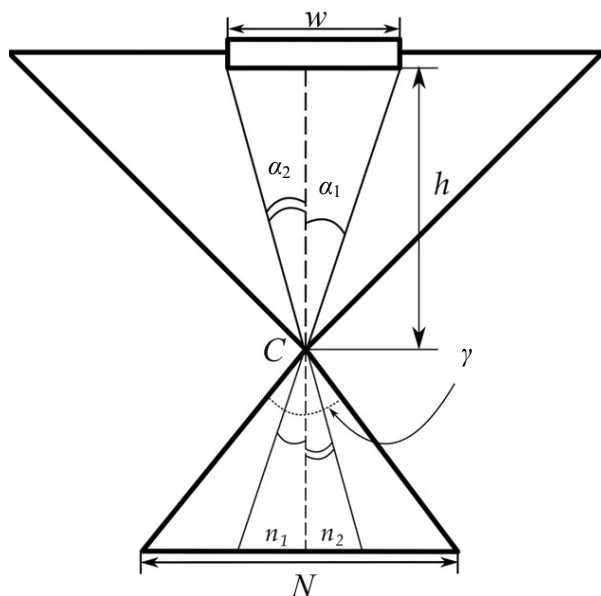
$$x_{dst} = H \cdot x_{src}, \quad (2)$$

где  $x_{src}$  — вектор, составленный из координат объекта на изображении,  $x_{dst}$  — вектор, составленный из координат объекта на наблюдаемой плоскости.

#### Алгоритм определения координат объекта на реальной плоскости

Для составления пар точек src–dst разработан алгоритм определения реальных координат объекта на наблюдаемой плоскости. В его основе лежит распознавание объекта с известными размерами на изображении и использование характеристик монокулярной системы. Для построения модели алгоритма воспользуемся схематическим представлением работы монокулярной камеры, изображенной на рис. 1.

Здесь камера соответствует нижнему треугольнику, основание которого является матрицей камеры с разрешением  $N$  пикселей по одной из осей и углом поля зрения  $\gamma$ . Верхний треугольник — зона видимости камеры, в которой на расстоянии  $h$  от камеры находится объект



**Рис. 1.** Геометрическая интерпретация работы камеры. Горизонтальная проекция.

Точка  $C$  — объектив камеры с разрешением  $N$  и углом поля зрения  $\gamma$ , прямоугольник с шириной  $w$  — объект интереса, удаленный от камеры на расстояние  $h$

с известным размером  $w$ . Искомое расстояние  $h$  можно найти по формуле:

$$h = \frac{w}{\tan \alpha_2 - \tan \alpha_1}, \quad (3)$$

где  $\alpha_1$  и  $\alpha_2$  — углы, соответствующие правой и левой границам ограничивающей рамки объекта. Положительное направление углов определяется в направлении против часовой стрелки относительно высоты  $h$ .

Для расчета удаленности  $h$  необходимо определить тангенсы характерных углов  $\alpha_1$  и  $\alpha_2$ . Для этого воспользуемся схемой рис. 1. Для произвольного номера пикселя верно:

$$\tan \alpha_i = \frac{n_i}{h_{cam}}, \quad i = 1, 2, \quad (4)$$

где  $h_{cam}$  — фокусное расстояние камеры. При этом для крайнего значения верно следующее:

$$\tan \frac{\gamma}{2} = \frac{N}{2h_{cam}}. \quad (5)$$

Тогда, выразив  $h_{cam}$  из (5) и подставив в (4), получаем

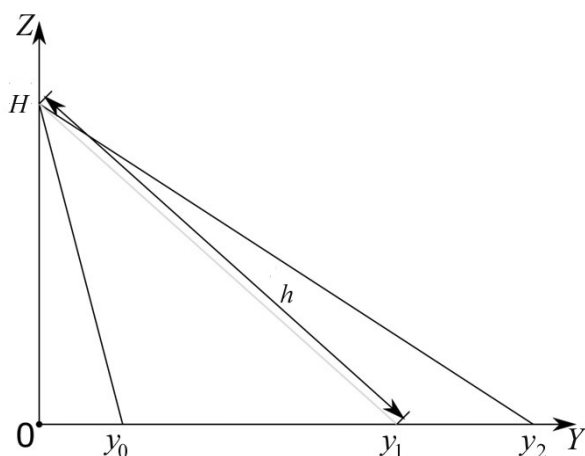
$$\tan \alpha_i = \frac{2n_i \tan \frac{\gamma}{2}}{N}, \quad i = 1, 2. \quad (6)$$

Использовать удаленность объекта от камеры в качестве одной из координат на реальной плоскости будет некорректно, т.к. данный отрезок не лежит в этой плоскости. Чтобы верно рассчитать координату, воспользуемся схемой работы стационарной монокулярной камеры наблюдения, изображенной на рис. 2.

Здесь  $H$  — высота установки камеры от наблюдаемой плоскости;  $h$  — удаленность объекта от камеры, рассчитанная по формуле (3);  $y_0$ ,  $y_2$  — координаты границ поля зрения монокулярной камеры;  $y_1$  — координата положения объекта интереса на наблюдаемой плоскости. По теореме Пифагора рассчитаем координату  $y_1$ :

$$y_1 = \sqrt{h^2 - H^2}. \quad (7)$$

Для определения горизонтальной координаты  $x$  объекта интереса на наблюдаемой плоскости определим по формуле (6) тангенс угла  $\alpha_x$  для соответствующего положения объекта.



**Рис. 2.** Схема работы монокулярной камеры. Вид сбоку.

Точка  $(0, H)$  — точка установки камеры,  $y_0$  и  $y_1$  — границы видимости камеры,  $y_1$  — координата, в которой находится объект интереса на расстоянии  $h$  от камеры

Тогда:

$$x = h \cdot \tan \alpha_x. \quad (8)$$

При этом условимся, что вертикальная линия, проходящая через центр изображения, соответствует  $x = 0$ , левая часть изображения —  $x < 0$ , правая —  $x > 0$ .

Следовательно, зная углы поля зрения камеры, разрешение изображения и высоту установки относительно наблюдаемой плоскости, можно определить координаты объекта интереса на данной плоскости, используя координаты объекта на изображении.

#### Итеративный подбор параметров для расчета матрицы гомографии

В предыдущем разделе описывается алгоритм определения координат объекта интереса на наблюдаемой плоскости. В качестве входных параметров алгоритма используются координаты объекта на изображении, высота установки камеры и ее угол поля зрения. Далее набор найденных пар координат используется для построения матрицы гомографии для области интереса в соответствии с алгоритмом, описанным в разделе "Матрица гомографии". Зачастую при работе с реальными данными с камер параметры, связанные с монокулярной системой, а именно высота установки и угол поля зрения, могут быть

неизвестны. В таком случае данные параметры могут быть подобраны итеративно.

Угол поля зрения камеры варьируется от  $20^\circ$  до  $130^\circ$  с шагом  $1^\circ$ , а высота установки — от 2 до 8 м с шагом 0.1 м. Далее выбирается пара значений угла и высоты, которые подставляются в формулы (6), (7) и (8) для перевода координат на изображении в координаты объекта на наблюдаемой плоскости. После расчета всех значений координат найденная пара точек используется для вычисления матрицы гомографии для данного угла поля зрения и высоты установки.

Для выбора итоговой матрицы гомографии используется информация об объекте с известными размерами, который используется для определения координат на наблюдаемой плоскости. В качестве критерия используется условие минимума средней ошибки определения размера объекта на изображении:

$$\varepsilon = \frac{1}{N} \sum_{i=0}^{N-1} |d_{\text{real}} - d_{\text{calc}}|, \quad \varepsilon \rightarrow \min, \quad (9)$$

где  $N$  — количество объектов;  $d_{\text{real}}$  — истинный размер данного объекта;  $d_{\text{calc}}$  — размер объекта, рассчитанный с помощью матрицы гомографии.

Таким образом можно подобрать параметры камеры, используя информацию о размерах объектов на ней. Данный алгоритм позволяет рассчитать матрицу преобразования для тех случаев, когда конфигурация монокулярной системы неизвестна.

#### Детектирование и трекинг

Для распознавания объекта на изображении используются алгоритмы на основе машинного обучения. Для того, чтобы разработанную систему можно было реализовать на маломощных вычислителях, таких как Jetson Orin Nano, в качестве детектора была выбрана облегченная архитектура YOLOv8-nano [16]. В отличие от других, данная архитектура обеспечивает высокую скорость обработки изображений в реальном времени при достаточной точности [16, 17].

В данной задаче экспериментальное исследование алгоритма проводилось в складском помещении, по которому передвигались погрузчики с типовым поддоном на вилах. В качестве объекта интереса выступал погрузчик, а объектом с известными размерами выступал поддон. Поэтому в рамках данного исследования для распознавания погрузчика на изображении был обучен детектор YOLOv8-nano, а для поиска поддона — сегментатор той же архитектуры и размера.

Для обучения детектора использовалось порядка 15 000 изображений с разрешением

1920 × 1080 пикселей. Данные изображения были получены с различных камер наблюдения, установленных в трех складских помещениях. Перед обучением с целью ускорить работу детектора разрешение входных изображений было уменьшено до 640 × 640 пикселей. Обучение длилось 50 эпох, данные были разделены на тренировочный и валидационный наборы в отношении 75 / 25. На рис. 3 изображен пример данных, которые были использованы для обучения детектора. Ограничивающие рамки отображают результат, выдаваемый детектором.

Для обучения сегментатора поддонов на вилах погрузчика применялись порядка 30 000 изображений, вырезанных с кадров в исходном разрешении, которые использовались для обу-

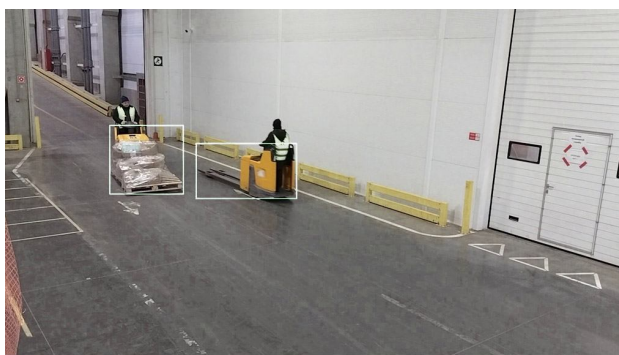


Рис. 3. Пример данных для обучения детектора.



Рис. 4. Пример данных для обучения сегментатора.

чения детектора. Размер данных изображений был предварительно изменен до 640 × 640 пикселей. Обучение длилось 30 эпох, данные были разделены на тренировочный и валидационный наборы в отношении 75 / 25. На рис. 4 изображен пример данных, которые были использованы для обучения. Зеленая область соответствует результату работы сегментатора.

Для расчета скорости движения объекта необходимо хранить историю его положений для каждого момента времени, когда он находится в поле зрения камеры. Для решения данной задачи требуется реализовать работу трекера. В настоящем исследовании использовалась технология BoT-SORT [18], модифицированная для решения поставленной задачи. Основная модификация заключается в введении зоны интереса, которая представляет собой выделенный на изображении многоугольник. Данная зона предназначена для фильтрации детекций удаленных и сложно различимых объектов. Все объекты, распознанные детектором и попадающие в данную зону, передаются трекеру для дальнейшей обработки. Детекции вне зоны интереса не используются в последующих расчетах. Данная модификация позволяет избежать ошибок, которые могут возникнуть при обработке объектов, занимающих слишком малую площадь на изображении или частично скрывающихся за краем изображения. Также в нашей реализации трекера используется фильтр Калмана, реализованный на языке C++ с целью ускорения работы.

#### *Выбор характерной точки объекта и уточнение полученных координат на изображении*

При выборе характерной точки для отслеживания важно учитывать перспективу, обусловленную положением камеры. Следует принимать во внимание возможные искажения и выбирать точки так, чтобы минимизировать их влияние при определении местоположения объекта интереса. Наиболее достоверная информация о положении движущегося объекта на наблюдаемой плоскости сосредоточена на нижней грани ограничивающей рамки. Можно выделить несколько ключевых точек на этом отрезке: середина, точки, связанные с объектом, вершины.

Наиболее простым решением является использование середины нижней грани в качестве точки отсчета. Однако этот подход имеет недостаток: позиция этой точки на наблюдаемой плоскости относительно объекта интереса будет изменяться в зависимости от того, где объект находится в кадре.

Другой вариант — это определение характерных точек на объекте, расположенных вблизи или на наблюдаемой плоскости. Этот метод основывается на том, что такие точки движутся вместе с объектом, учитывая перспективу камеры. Примером характерных точек могут служить колеса автомобиля. Недостаток этого метода заключается в том, что часто трудно найти точку, находящуюся близко к наблюдаемой плоскости и всегда видимую камерой.

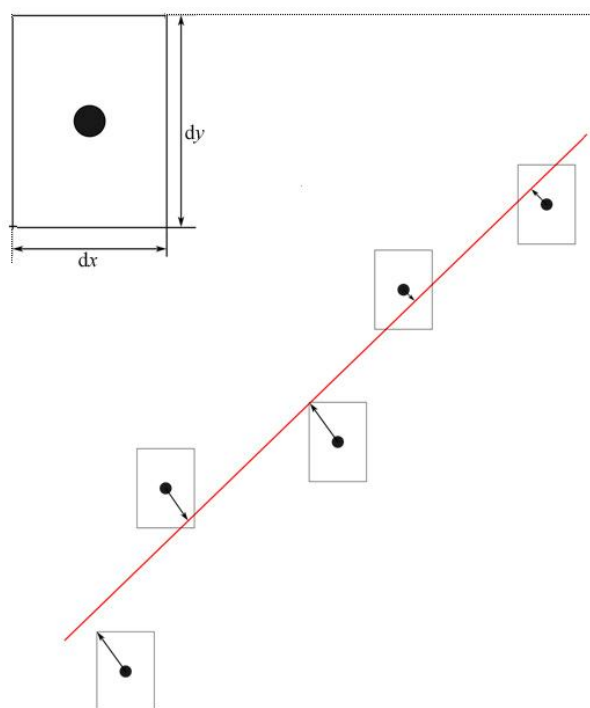
Последнее предложенное решение заключается в использовании дальнего угла ограничивающей рамки относительно камеры. Перспективные искажения при этом варианте оказывают минимальное влияние на определение положения объекта. Чтобы точно определить положение дальнего угла, можно учитывать положение камеры при ее калибровке или использовать информацию о траектории движения объекта. Например, если движение начинается из левого верхнего угла и заканчивается в правом нижнем, то дальним углом будет правый нижний угол рамки.

Для повышения точности полученных координат разработан алгоритм, который снижает случайные погрешности, возникающие при использовании методов глубокого обучения. Основой метода является усреднение значений координат на основе набора точек. Вначале задается размер окна для корректировки. Затем данный массив заполняется точками в течение нескольких кадров. Как только накоплено достаточное количество точек, по ним строится линейная регрессия, и координаты корректируются ближе к этой линии, не выходя за пределы совокупной средней погрешности. Величина погрешности определяется, исходя из ошибки используемых методов распознавания объектов на изображении.

На рис. 5 показана схема работы алгоритма, где прямоугольники обозначают максимальную коррекцию по каждой оси, а прямая линия — линейная регрессия, построенная по координатам на изображении. Алгоритм состоит из следующих шагов:

- определение размера окна для усреднения;
- построение линейной регрессии по всем точкам в окне;
- коррекция координат на основе полученной регрессии.

Использование данного усреднения позволяет уменьшить влияние погрешности используемых методов для распознавания объектов интереса на изображении. Ограничением данного подхода является требование, что траектория движения объектов в кадре должна быть линейной или изменяться достаточно плавно. Как показывает



**Рис. 5.** Коррекция точек расчета.

Окружности — детекции объекта на плоскости изображения, прямоугольники — максимально допустимая коррекция, прямая линия — линейная регрессия

практика, данное ограничение не является жестким. Зачастую инерционность объектов интереса, например погрузчиков или автомобилей, не позволяет им двигаться по неподходящим для данного алгоритма траекториям.

### Расчет скорости

После расчета траектории объекта на наблюдаемой плоскости и при наличии соответствующих временных меток можно рассчитать его среднюю скорость. Для этого необходимо разделить перемещение объекта между двумя точками на время, за которое он преодолел данное расстояние:

$$v_{\text{avg}} = \frac{|P_0 - P_1|}{\Delta t}, \quad (10)$$

где  $P_0, P_1$  — координаты двух точек траектории объекта на наблюдаемой плоскости;  $\Delta t$  — время, за которое объект переместился от точки  $P_0$  до точки  $P_1$ .

Таким образом, для расчета средней скорости движения объекта необходимо хранить историю его положений во времени. Для решения данной задачи используется трекер объектов.

## РЕЗУЛЬТАТЫ

### Условия эксперимента

Для проверки работоспособности алгоритма и расчета допустимых ограничений была проведена серия экспериментов. В качестве входных данных рассматривалось движение погрузчиков с поддоном на вилах по складскому помещению. Была записана серия видеозаписей проездов. В качестве монокулярной системы использовалась конфигурация с параметрами, указанными в табл. 1.

Полученные видеозаписи были обработаны с применением алгоритма, описанного выше. Было обработано порядка 2 ч видео в реальном времени, которые включают в себя 150 проездов погрузчиков в поле зрения камеры. Для обработки была предварительно создана зона интереса. Данная зона изображена на рис. 6 и соответствует площади, ограниченной белым многоугольником.

Табл. 1. Параметры монокулярной системы

|                                         |                    |
|-----------------------------------------|--------------------|
| Высота установки ( $H$ )                | 3 м                |
| Угол поля зрения ( $\gamma$ )           | $60^\circ$         |
| Разрешение изображения ( $M \times N$ ) | $1920 \times 1080$ |

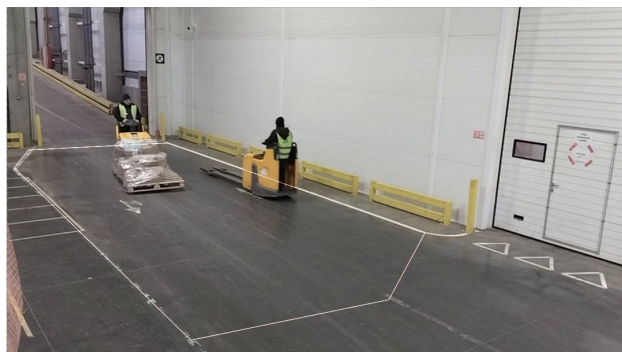


Рис. 6. Зона интереса.

Для всех погрузчиков с поддоном на вилах, попавших в данную область, распознавалась маска передней части поддона. Далее маска использовалась для расчета положения погрузчика при использовании алгоритма, представленного в разделе "Описание алгоритма". В качестве наблюдаемой плоскости в данном случае выступала плоскость пола, по которой двигались погрузчики.

### Расчет матрицы гомографии

В данном эксперименте был протестирован алгоритм автоматической калибровки камеры путем итерационного подбора параметров, описанного в разделе "Итеративный подбор параметров для расчета матрицы гомографии". Для проведения данного теста мы считаем, что параметры системы (кроме разрешения изображения) заранее неизвестны. Распределение рассчитанной ошибки в зависимости от параметров камеры изображено на рис. 7.

Отсутствие части значений на данном распределении (белая область) связано с тем, что при использовании параметров из этой области полученное расстояние до объекта становилось меньше либо равным высоте установки камеры, чего не может быть.

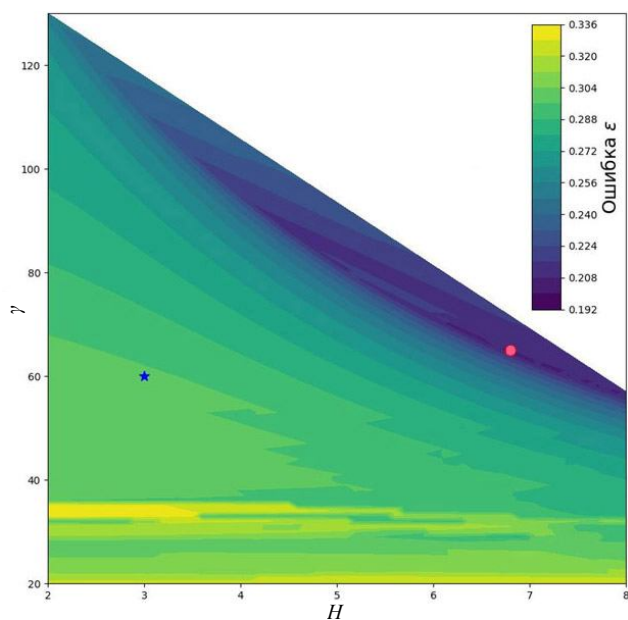


Рис. 7. Распределение рассчитанной ошибки при автоматической калибровке.

При этом параметры, которые удовлетворяют критерию (9), с ошибкой  $\varepsilon = 0.197$ , соответствуют следующим значениям:

$$\gamma = 65^\circ,$$

$$H = 6.8 \text{ м.}$$

Данные значения отмечены точкой на рис. 7. Звезда соответствует истинным параметрам системы, для них значение функции ошибки  $\varepsilon = 0.297$ .

Подобное отличие рассчитанных параметров камеры от истинных может быть связано со схожестью изображений в зависимости от положения камеры. На рис. 8 схематично изображен описываемый эффект. При данных конфигурациях изображения не будут сильно отличаться друг от друга. Подобный эффект возможен при малых изменениях угла поля зрения  $\gamma$  и  $\gamma'$  при сравнении с высотой  $H$  и  $H'$ . Иными словами, изменение угла поля зрения камеры вносит значительно больший вклад в итоговый результат рассчитанных координат, чем изменение высоты установки. Данное утверждение согласуется с параметрами, полученными в ходе автоматической калибровки. Угол поля зрения  $\gamma$  отличается на  $5^\circ$ , что составляет 8% от истинного, а высота установки на 3.8 м (227%).

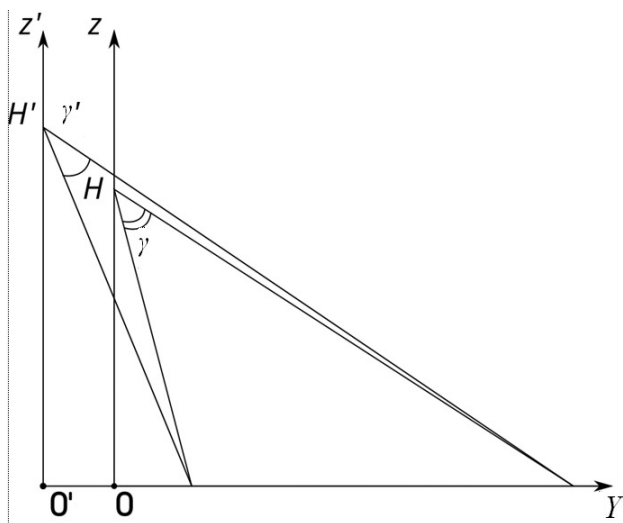


Рис. 8. Разница параметров монокулярной системы.

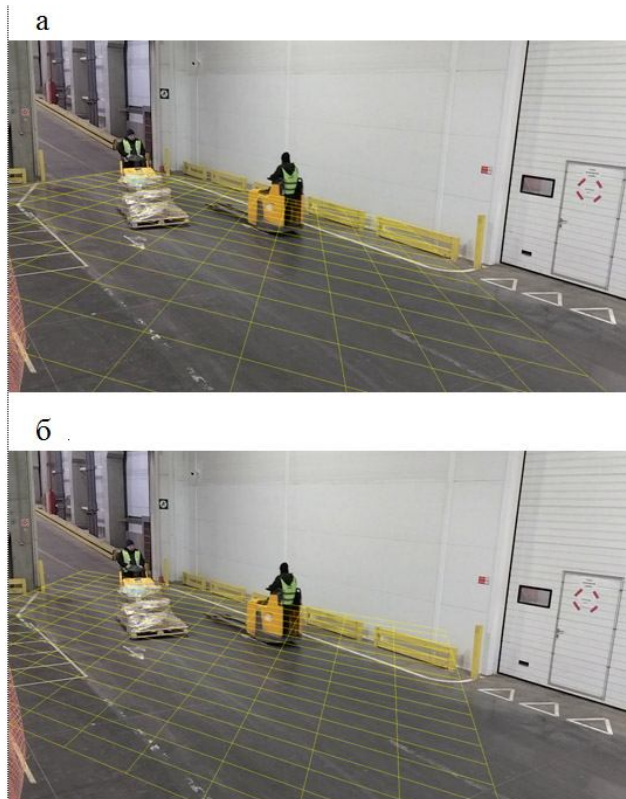


Рис. 9. Преобразование гомографии.  
а — истинные параметры системы,  
б — алгоритм итеративного подбора

Далее рассчитаем матрицу гомографии для истинных, указанных в табл. 1, и рассчитанных параметров монокулярной системы. На рис. 9, а, изображена координатная сетка, полученная в результате преобразования (2).

Данная сетка отображает перспективные искажения, вносимые положением камеры. Каждая ячейка обладает размерами  $1 \times 1$  м. Заметно, как площадь ячейки уменьшается при увеличении удаленности объекта от камеры вдоль наблюдаемой плоскости. Сетка, отображающая полученную матрицу преобразования для параметров, полученных алгоритмом автоматической калибровки, изображена на рис. 9, б. Данная сетка является более сжатой по сравнению с той, которая получилась при использовании истинных параметров. Далее рассмотрим влияние выбора рассчитанных матриц при вычислении средней скорости движения погрузчиков.

### Расчет средней скорости

#### Влияние расчета матрицы гомографии

Был проведен эксперимент по вычислению средней скорости проезжающих погрузчиков в поле зрения камеры. Для расчета использовались матрицы гомографии, полученные с использованием истинных параметров монокулярной камеры и с помощью алгоритма автоматической калибровки. В эксперименте скорость погрузчиков, рассчитанная с помощью предложенного алгоритма, сравнивалась со скоростью, полученной с помощью GPS. В качестве результата алгоритма бралось значение, полученное на момент выезда погрузчика из зоны интереса.

На рис. 10 изображено распределение ошибок расчета скорости в зависимости от используемого метода построения матрицы гомографии. Сплошные колонки отвечают за скорости, полученные с использованием истинных параметров камеры, штриховые — при помощи автокалибровки. В качестве характерных точек бралась дальняя от камеры вершина ограничивающей рамки погрузчика. При этом полученные точки корректировались в соответствии с алгоритмом, описанным в разделе "Выбор характерной точки и уточнение полученных координат на изображении". Заметно, что дисперсия значений, полученных с помощью автоматической калибровки, превышает значения, полученные с помощью истинных параметров камеры. В табл. 2 представлена доля рассчитанных

**Табл. 2.** Влияние алгоритма построения матрицы гомографии на точность расчета средней скорости

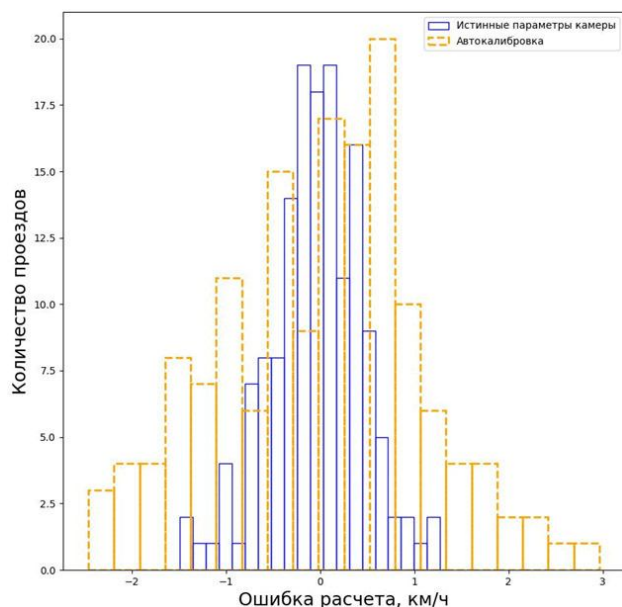
| Скорости   | Истинные параметры, % | Автокалибровка, % |
|------------|-----------------------|-------------------|
| < 0.5 км/ч | 0.87                  | 0.67              |
| < 1 км/ч   | 0.98                  | 0.84              |
| < 1.5 км/ч | 1                     | 0.95              |
| < 2 км/ч   | 1                     | 0.97              |

скоростей, ошибка которых не превышает определенный порог.

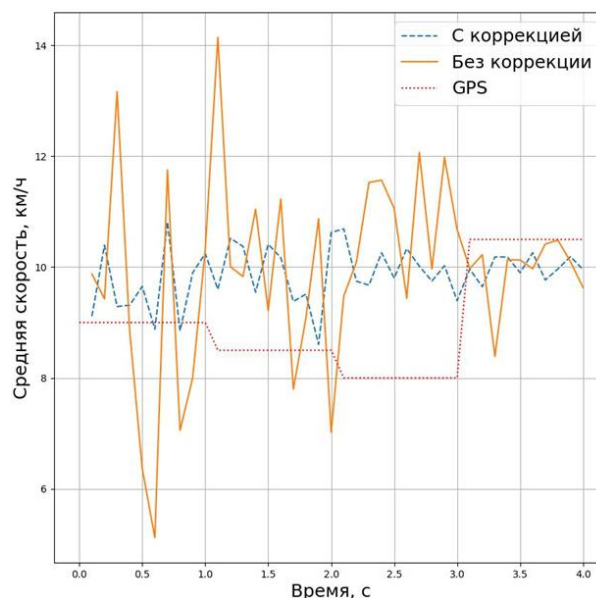
При использовании истинных параметров монокулярной системы 98% рассчитанных скоростей попадают в пределы погрешности  $\pm 1$  км/ч, а использование матрицы, полученной с помощью автокалибровки, позволяет достичь метрики в 84%. Следовательно, использование алгоритма автоматической калибровки значительно влияет на точность расчета скорости, однако он все еще может быть применен в ситуациях, не требующих сильных ограничений, и когда параметры системы неизвестны.

#### Влияние корректировки характерных точек

Рассмотрим влияние алгоритма корректировки характерных точек на расчет скорости. Для расчета матрицы гомографии будем использовать истинные параметры системы. В качестве



**Рис. 10.** Распределение GPS-скоростей.



**Рис. 11.** Коррекция точек расчета при вычислении скорости.

характерной точки будем использовать дальний нижний угол ограничивающей рамки. На рис. 11 изображен результат расчета средней скорости. Сплошная линия соответствует расчету без использования коррекции характерных точек, штриховая — с коррекцией, пунктирная — скорость по GPS.

В данном случае значение средней скорости в каждый момент времени рассчитывается между текущим положением погрузчика на кадре и начальным положением, полученным в момент въезда в зону интереса. Скорость, рассчитанная с использованием алгоритма коррекции, получается более гладкой по сравнению со скоростью, рассчитанной без коррекции.

В табл. 3 представлена доля рассчитанных скоростей, ошибка которых не превышает определенный порог, в зависимости от применения алгоритма коррекции характерных точек.

При использовании алгоритма коррекции удается увеличить точность расчета средней скорости движения погрузчиков. Например, для порога  $\pm 1$  км/ч коррекция точек позволяет увеличить долю правильных ответов с 81% до 98%. В результате коррекции точек удастся уменьшить влияние случайной ошибки, вызванной алгоритмами распознавания погрузчиков и фронтальной части поддона на изображении.

#### Влияние выбора характерных точек

Рассмотрим влияние выбора характерной точки на расчет средней скорости движения погрузчиков в зоне интереса. В качестве матрицы гомографии использовалась матрица, полученная с использованием истинных параметров системы. При этом характерные точки корректировались в соответствии с алгоритмом, описанным в разделе "Выбор характерной точки объекта и уточнение полученных координат на изображении". В качестве характерных точек были выбраны: центр нижней грани ограничивающей рамки, центр фронтальной части поддона и дальний нижний угол относительно камеры угол ограничивающей рамки, в данном

**Табл. 3.** Влияние коррекции характерных точек на расчет средней скорости

| Скорости   | С коррекцией, % | Без коррекции, % |
|------------|-----------------|------------------|
| < 0.5 км/ч | 0.87            | 0.73             |
| < 1 км/ч   | 0.98            | 0.81             |
| < 1.5 км/ч | 1               | 0.9              |
| < 2 км/ч   | 1               | 0.94             |

**Табл. 4.** Метрика отклонения средней скорости в зависимости от характерных точек

| Скорости   | Дальний угол, % | Центр поддона, % | Центр нижней грани, % |
|------------|-----------------|------------------|-----------------------|
| < 0.5 км/ч | 0.87            | 0.80             | 0.75                  |
| < 1 км/ч   | 0.98            | 0.93             | 0.90                  |
| < 1.5 км/ч | 1               | 0.96             | 0.95                  |
| < 2 км/ч   | 1               | 0.99             | 0.99                  |

случае это правая нижняя вершина. В табл. 4. представлена доля рассчитанных скоростей, ошибка которых не превышает определенный порог, в зависимости от выбранной характерной точки.

Как и ожидалось, наименьшую точность показал расчет для точек, находящихся на центре нижней грани ограничивающей рамки. Наибольшее количество рассчитанных скоростей при интервале погрешности в  $\pm 1$  км/ч было достигнуто при использовании дальнего угла рамки и составило 98%. Для более слабых ограничений разница становится меньше.

#### ЗАКЛЮЧЕНИЕ

В работе был предложен алгоритм для расчета положений и скоростей движения объектов на плоскости по данным с монокулярной камеры. Основой алгоритма является построение матрицы преобразования точек на плоскости изображения в точки на наблюдаемой плоскости, по которой перемещаются объекты в поле зрения камеры. Для построения матрицы используется информация о реальном размере объекта и параметры монокулярной системы — высота установки и угол поля зрения. Также разработан алгоритм автоматической калибровки камеры. В качестве методов распознавания объектов на изображении использовались нейросетевые детектор и сегментатор. Для уменьшения ошибки перспективы проведено исследование по выбору характерной точки объекта на изображении. Чтобы уменьшить погрешность нейросетевых методов распознавания, разработан алгоритм корректировки характерных точек на основе усреднения значений координат. Проведен эксперимент с определением средней скорости движения погрузчиков по распределительному складу в поле зрения монокулярной камеры. Результаты эксперимента показали, что 98% рассчитанных средних

скоростей лежат в интервале  $\pm 1$  км/ч от эталонных значений.

#### СПИСОК ИСТОЧНИКОВ

1. *Vodichev N. et al.* Real-time distance estimation algorithm for objects in warehouse based on monocular camera data for an autonomous unmanned forklift // The European Physical Journal Special Topics. Springer, 2025. P. 1–13. DOI: 10.1140/epjs/s11734-024-01452-4
2. *Hua S., Kapoor M., Anastasiu D.C.* Vehicle tracking and speed estimation from traffic videos // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018. P. 153–160. DOI: 10.1109/CVPRW.2018.00028
3. *Fernández L.D., Hernández M.A., Garcí aDI.* Vision-based vehicle speed estimation: A survey // IET Intelligent Transport Systems. Wiley Online Library, 2021. Vol. 15, no. 8. P. 987–1005. DOI: 10.1049/itr2.12079
4. *Маликова Т., Соловьева Е., Радочинская А.* Автоматизация рабочего цикла погрузчика на механизированном контейнерном терминале // АЭРОКОСМИЧЕСКОЕ ПРИБОРОСТРОЕНИЕ И ЭКСПЛУАТАЦИОННЫЕ ТЕХНОЛОГИИ. Четвертая Международная научная конференция. Том Часть 1. Санкт-Петербург, 2023. С. 220–224. URL: <https://elibrary.ru/item.asp?id=53731491&pff=1>
5. *Wang F. et al.* Efficient stereo visual simultaneous localization and mapping for an autonomous unmanned forklift in an unstructured warehouse // Applied Sciences. MDPI, 2020. Vol. 10, no. 2. P. 698. DOI: 10.3390/app10020698
6. *Jung E.-J. et al.* Localization for an unmanned forklift in a refrigerated warehouse // 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE, 2016. P. 53–55. DOI: 10.1109/URAI.2016.7734019
7. *LeCun Y., Bengio Y., Hinton G.* Deep learning // Nature. Nature Publishing Group UK London, 2015. Vol. 521, no. 7553. P. 436–444. DOI: 10.1038/nature14539
8. *Dubská M. et al.* Fully automatic roadside camera calibration for traffic surveillance // IEEE Transactions on Intelligent Transportation Systems. IEEE, 2014. Vol. 16, no. 3. P. 1162–1171. DOI: 10.1109/TITS.2014.2352854
9. *Dubská M., Herout A., Sochor J.* Automatic camera calibration for traffic understanding // BMVC. 2014. Vol. 4. P. 8. DOI: 10.5244/C.28.42
10. *Sochor J., Juránek R., Herout A.* Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement // Computer Vision and Image Understanding. Elsevier, 2017. Vol. 161. P. 87–98. DOI: 10.1016/j.cviu.2017.05.015
11. *Luvizon D.C., Nassu B.T., Minetto R.* A video-based system for vehicle speed measurement in urban roadways // IEEE Transactions on Intelligent Transportation Systems. IEEE, 2016. Vol. 18, no. 6. P. 1393–1404. DOI: 10.1109/TITS.2016.2606369
12. *Gretns A., Varkentin V., Goryaev N.* Determining vehicle speed based on video using convolutional neural network // Transportation Research Procedia. Elsevier, 2020. Vol. 50. P. 192–200. DOI: 10.1016/j.trpro.2020.10.024
13. *Vakili E., Shoaran M., Sarmadi M.R.* Single-camera vehicle speed measurement using the geometry of the imaging system // Multimedia Tools and Applications. Springer, 2020. Vol. 79, no. 27. P. 19307–19327. DOI: 10.1007/s11042-020-08761-5
14. *Neamah S.B., Karim A.A.* Real-time traffic monitoring system based on deep learning and yolov8 // Aro-the scientific journal of koya university. 2023. Vol. 11, no. 2. P. 137–150. DOI: 10.14500/aro.11327
15. *Dubrofsky E.* Homography estimation // Diplomová práce. Vancouver: Univerzita Britské Kolumbie. 2009. Vol. 5.
16. Ultralytics YOLO11 [Электронный ресурс]. 2023. URL: <https://github.com/ultralytics/ultralytics> (дата обращения: 18.03.2024).
17. *Terven J., Córdova-Esparza D.-M., Romero-González J.-A.* A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas // Machine learning and knowledge extraction. MDPI, 2023. Vol. 5, no. 4. P. 1680–1716. DOI: 10.3390/make5040083
18. *Aharon N., Orfaig R., Bobrovsky B.-Z.* Bot-sort: Robust associations multi-pedestrian tracking // arXiv preprint. 2022. Id. arXiv:2206.14651. DOI: 10.48550/arXiv.2206.14651

**Московский физико-технический институт,  
г. Долгопрудный**

Контакты: Водичев Никита Алексеевич,  
vodichev.na@phystech.edu

Материал поступил в редакцию 20.03.2025

# ALGORITHM FOR CALCULATING THE POSITION AND SPEED OF OBJECTS ON A PLANE USING DATA FROM A MONOCULAR CAMERA

N. A. Vodichev, D. A. Gavrilov, A. V. Leus, V. A. Efremov, I. V. Kholodnyak, V. A. Zuev,  
M. M. Parshikov, A. Yu. Troegubov, N. V. Gershtein, V. E. Laukhin, A. E. Antipov

*Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region, Russia*

This article presents an algorithm for the automatic calculation of a homography matrix, intended for estimating the speed of forklifts in warehouse conditions. The proposed algorithm uses video streams from surveillance cameras to transform object coordinates from a perspective image into a metric space, enabling the determination of forklift positions in real measurement units. The algorithm implements automatic calibration based on the recognition of pallets with known dimensions and the use of camera characteristics. An automatic parameter selection algorithm is proposed for the monocular system in cases where these parameters are not known in advance. An experiment using real video recordings demonstrates the applicability of the proposed algorithm for monitoring logistics processes and improving the efficiency of warehouse transport management. The experiment showed that when calculating average forklift speed, 98% of the obtained values fall within an error margin of  $\pm 1$  km/h.

**Keywords:** speed calculation, computer vision, neural networks, homography matrix

## REFERENCES

- Vodichev N. et al. Real-time distance estimation algorithm for objects in warehouse based on monocular camera data for an autonomous unmanned forklift. *The European Physical Journal Special Topics*. Springer, 2025. P. 1–13. DOI: 10.1140/epjs/s11734-024-01452-4
- Hua S., Kapoor M., Anastasiu D.C. Vehicle tracking and speed estimation from traffic videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. USA: Salt Lake City, 2018. P. 153–160. DOI: 10.1109/CVPRW.2018.00028
- Fernández L.D., Hernández M.A., García D.I. Vision-based vehicle speed estimation: A survey. *IET Intelligent Transport Systems*. Wiley Online Library, 2021. Vol. 15, no. 8, pp. 987–1005. DOI: 10.1049/itr2.12079
- Malikova T.E., Soloveva E.E., Radochinskaia A.Zh. [Loader work cycle automation at the mechanized container terminal]. *IV Mezhdunarodnaya nauchnaya konferentsiya: "Aerokosmicheskoe priborostroenie i ekspluatatsionnye tekhnologii". T. 1* [Proc. IV Int. Sci. Conf.: "Aero-space instrumentation and operational technologies". Vol. 1]. Saint Petersburg, 2023. P. 220–224. (In Russ.). URL: <https://elibrary.ru/item.asp?id=53731491&pf=1>
- Wang F. et al. Efficient stereo visual simultaneous localization and mapping for an autonomous unmanned forklift in an unstructured warehouse. *Applied Sciences*. MDPI, 2020, vol. 10, no. 2, Id. 698. DOI: 10.3390/app10020698
- Jung E.-J. et al. Localization for an unmanned forklift in a refrigerated warehouse. *13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2016. P. 53–55. DOI: 10.1109/URAI.2016.7734019
- LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*, 2015, vol. 521, no. 7553, pp. 436–444. DOI: 10.1038/nature14539
- Dubská M. et al. Fully automatic roadside camera calibration for traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*. IEEE, 2014, vol. 16, no. 3, pp. 1162–1171. DOI: 10.1109/TITS.2014.2352854
- Dubská M., Herout A., Sochor J. Automatic camera calibration for traffic understanding. *BMVC*, 2014, vol. 4, pp. 8. DOI: 10.5244/C.28.42
- Sochor J., Juránek R., Herout A. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding*, 2017, vol. 161, pp. 87–98. DOI: 10.1016/j.cviu.2017.05.015
- Luvizon D.C., Nassu B.T., Minetto R. A video-based system for vehicle speed measurement in urban roadways. *IEEE Transactions on Intelligent Transportation Systems*. IEEE, 2016, vol. 18, no. 6, pp. 1393–1404. DOI: 10.1109/TITS.2016.2606369
- Grents A., Varkentin V., Goryaev N. Determining vehicle speed based on video using convolutional neural network. *Transportation Research Procedia*, 2020, vol. 50, pp. 192–200. DOI: 10.1016/j.trpro.2020.10.024
- Vakili E., Shoran M., Sarmadi M.R. Single-camera vehicle speed measurement using the geometry of the imaging system. *Multimedia Tools and Applications*, 2020, vol. 79, no. 27, pp. 19307–19327. DOI: 10.1007/s11042-020-08761-5
- Neamah S.B., Karim A.A. Real-time traffic monitoring system based on deep learning and yolov8. *Aro-the scientific journal of Koya University*, 2023, vol. 11, no. 2, pp. 137–150. DOI: 10.14500/aro.11327

15. Dubrofsky E. *Homography estimation*. Diplomová práce. Vancouver: Univerzita Britské Kolumbie, 2009. Vol. 5. 32 p.
16. *Ultralytics YOLO11*. 2023.  
URL: <https://github.com/ultralytics/ultralytics> (accessed: 18.03.2024).
17. Terven J., Córdova-Esparza D.-M., Romero-González J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 2023, vol. 5, no. 4, pp. 1680–1716. DOI: 10.3390/make5040083
18. Aharon N., Orfaig R., Bobrovsky B.-Z. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint*, 2022, Id. arXiv:2206.14651.  
DOI: 10.48550/arXiv.2206.14651

Contacts: *Vodichev Nikita Alekseevich*,  
vodichev.na@phystech.edu

Article received by the editorial office on 20.03.2025

## INTRODUCTION

Computer vision technologies are actively used in various fields — from autonomous vehicles to security systems. Such systems are used to monitor pedestrian areas, public roads, warehouses, and in the control of unmanned vehicles. These technologies are mainly used in the field of environmental analysis for real-time decision-making. In particular, such systems can evaluate the trajectories of objects, determine their speed and position, and predict potential collisions or violations of rules [1–4]. These functions are especially important in highly dynamic and space-constrained conditions, for example, at warehouse sites, where it is necessary to ensure safe interaction between forklifts, workers, and other infrastructure elements.

Often, forklift-mounted equipment is used to solve movement safety problems. In particular, the problem of collision prevention is solved with the help of parking sensors, lidars, or radars [5, 6]. However, the use of these technologies has a number of disadvantages and limitations:

- the equipment should be installed on each forklift, and the consequence is the high cost of purchase and installation;
- performing a limited list of tasks — measuring distances to objects and preventing collisions;
- difficulties with scaling.

Nevertheless, most modern warehouses already have installed video surveillance systems, which can be effectively used to solve security problems. The introduction of computer vision algorithms opens up new opportunities for monitoring and preventing collisions. However, a large number of installed cameras and the differences between warehouses lead to a significant increase in the volume and variety of data that must be processed. In such conditions, automated analysis methods that can effectively process large amounts of information and be stable become relevant.

As such a method, one can use artificial neural networks, which have proven their effectiveness in solving a wide range of problems [7]. The generalizing ability of neural networks and the ease of installing monocular cameras make it possible to simplify the implementation of the developed solution in existing warehouses. In combination with the available forklift technologies, the proposed algorithm will expand the list of tasks performed. Moreover, the use of this system allows for the reduction of costs, eliminating the need for significant investments in new equipment.

This paper proposes an algorithm for calculating the positions and speeds of objects on a plane based on data from a monocular camera. Convolutional neural networks are used to detect objects of interest. The main feature of the proposed algorithm is the automatic calculation of the homography matrix, which eliminates the need for preliminary manual calibration of the camera and the use of external information about the observed plane. The homography matrix is compiled using the recognition of objects with known dimensions associated with the observed plane. The main advantages of the algorithm are:

- use of a monocular system instead of radars and stereo cameras;
- the ability to process large volumes of video data and streams from various cameras;
- automatic calculation of the homography matrix to determine the positions of objects on a plane;
- 98% of the calculated average speeds lie within 1 km/h of the reference values.

## LITERATURE REVIEW

Many methods have been developed to determine the position and speed of objects using data from a monocular camera. The works [8–10] present a method for calculating the speed of vehicles using three vanishing points. To determine the metric characteris-

tics of the resulting grid, it is necessary to calculate the scale factor. In the work [9], statistics on the average dimensions of cars are used for this purpose, which are compared with the dimensions in pixels. This can cause errors associated with the diversity of data in the statistics and the position of the camera. In the article [10], instead of using statistical data, the authors propose to classify the make, model, and year of manufacture of the car and take into account its actual dimensions. This method allows for increasing the accuracy of calculating the scale factor but is difficult to implement since a large amount of data is required to train the classifier.

The scientific literature also presents methods that use preliminary calibration of the camera to determine the location and the distance traveled based on meta-information about the observed part of the road [11, 12]. The disadvantage of this approach is the requirement for the availability of precise data on the observed plane, which is not always possible.

In addition, scientists propose algorithms that allow determining the distance of objects from the camera based on the recognition of objects with pre-known dimensions. For example, in the article [13], an algorithm using a car license plate as an object with known dimensions is proposed. The basis of the algorithm is a geometric model of the system, and the change in the area of the license plate in the image is used to calculate the speed. The main disadvantage of this approach is the need to recognize the number in each frame. This can be difficult, since the number may be dirty or not visible, and its contours are likely to be blurred at high speeds and in insufficient lighting. In the work [14], segmentation of vehicles and two zones of interest are used. For each zone, a coefficient  $PPM$ , is calculated based on the found mask and the true dimensions of the car and determines the number of pixels in one meter. Then this coefficient is used to calculate the distance traveled. The disadvantage of this approach is the preliminary calibration and calculation of the coefficient  $PPM$ , on which the accuracy of calculating the speed of objects directly depends.

## ALGORITHM DESCRIPTION

### *Camera calibration*

#### **Homography matrix**

To determine the position of the detected object on the observed plane, it is necessary to build a correspondence between the coordinates in the image and the coordinates on this plane. For this, you can use the homography matrix [15]:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}. \quad (1)$$

Expression (1) is a matrix  $3 \times 3$ , in which the components  $h_{11}$ ,  $h_{12}$ ,  $h_{21}$  and  $h_{22}$  are responsible for the rotation and stretching of the coordinate system,  $h_{13}$  and  $h_{23}$  are responsible for the shift of the origin, and  $h_{31}$  and  $h_{32}$  are responsible for perspective distortion. The conversion of a vector from one coordinate system to another is carried out by multiplying by the homography matrix:

$$x_{dst} = H \cdot x_{src}, \quad (2)$$

where  $x_{src}$  is a vector composed of the coordinates of the object in the image,  $x_{dst}$  is a vector composed of the coordinates of the object on the observed plane.

#### **Algorithm for determining the coordinates of an object on a real plane**

To create pairs of src–dst points, an algorithm for determining the real coordinates of an object on the observed plane has been developed. It is based on recognizing an object with known dimensions in the image and using the characteristics of a monocular system. To build a model of the algorithm, we will use the schematic representation of the monocular camera shown in Fig. 1.

**Fig. 1.** Geometric interpretation of camera operation. Horizontal projection. Point  $C$  — camera lens with resolution  $N$  and field of view angle  $\gamma$ ; rectangle with width  $w$  — object of interest at a distance  $h$  from the camera

Here, the camera corresponds to the lower triangle, the base of which is the camera matrix with a pixel resolution  $N$  along one of the axes and a field of view angle  $\gamma$ . The upper triangle is the camera's visibility zone, in which an object with a known size  $w$  is located at a distance  $h$  from the camera. The desired distance  $h$  can be found using the formula:

$$h = \frac{w}{\tan \alpha_2 - \tan \alpha_1}, \quad (3)$$

where  $\alpha_1$  and  $\alpha_2$  are the angles corresponding to the right and left boundaries of the object's bounding box. The positive direction of the angles is determined counterclockwise relative to the height  $h$ .

To calculate the distance  $h$ , it is necessary to determine the tangents of the characteristic angles  $\alpha_1$  and  $\alpha_2$ . For this, we will use the diagram in Fig. 1. For an arbitrary pixel number, the following is true:

$$\tan \alpha_i = \frac{n_i}{h_{\text{cam}}}, \quad i = 1, 2, \quad (4)$$

where  $h_{\text{cam}}$  is the focal length of the camera. In this case, for the extreme value, the following is true:

$$\tan \frac{\gamma}{2} = \frac{N}{2h_{\text{cam}}}. \quad (5)$$

Then, expressing  $h_{\text{cam}}$  from (5) and substituting into (4), we obtain

$$\tan \alpha_i = \frac{2n_i \tan \frac{\gamma}{2}}{N}, \quad i = 1, 2. \quad (6)$$

It will be incorrect to use the distance of the object from the camera as one of the coordinates on the real plane, since this segment does not lie in this plane. To correctly calculate the coordinate, we will use the operation diagram of a stationary monocular surveillance camera, shown in Fig. 2.

Here  $H$  is the height of the camera installation relative to the observed plane;  $h$  is the distance of the object from the camera, calculated using formula (3);  $y_0$ ,  $y_2$  are the coordinates of the boundaries of the field of view of the monocular camera;  $y_1$  is the coordinate of the position of the object of interest on the observed plane.

**Fig. 2.** Scheme of operation of monocular camera. Side view.

Point  $(0, H)$  — camera installation point;  $y_0$  and  $y_1$  — camera vision limits;  $y_1$  — coordinate, where the object of interest is located at a distance  $h$  from the camera

Using the Pythagorean theorem, we calculate the coordinate  $y_1$ :

$$y_1 = \sqrt{h^2 - H^2}. \quad (7)$$

To determine the horizontal coordinate  $x$  of the object of interest on the observed plane, we define the tangent of the angle  $\alpha_x$  for the corresponding position of the object using formula (6).

Then:

$$x = h \cdot \tan \alpha_x. \quad (8)$$

In this case, we agree that the vertical line passing through the center of the image is equal to  $x = 0$ , the left part of the image —  $x < 0$ , the right —  $x > 0$ .

Therefore, knowing the angles of the camera's field of view, the image resolution, and the installation height relative to the observed plane, we can determine the coordinates of the object of interest on this plane using the coordinates of the object in the image.

#### Iterative selection of parameters for calculating the homography matrix

The previous section describes the algorithm for determining the coordinates of the object of interest on the observed plane. The coordinates of the object in the image, the camera installation height, and its angle of field of view are used as input parameters of the algorithm. Next, the set of found coordinate pairs is used to construct the homography matrix for the region of interest in accordance with the algorithm described in the Homography Matrix chapter. Often, when working with real data from cameras, the parameters associated with the monocular system, namely the installation height and the field of view angle, may be unknown. In this case, these parameters can be selected iteratively.

The camera field of view angle varies from  $20^\circ$  to  $130^\circ$  with a step of  $1^\circ$ , and the installation height — from 2 to 8 m with a step of 0.1 m. Next, a pair of angle and height values are selected and substituted into formulas (6), (7) and (8) to convert the coordinates in the image into the coordinates of the object on the observed plane. After calculating all the coordinate values, the found pair of points is used to calculate the homography matrix for a given field of view angle and installation height.

To select the final homography matrix, we use information about an object with known dimensions to determine the coordinates on the observed plane. The criterion is the condition of the minimum average error in determining the size of an object in the image:

$$\varepsilon = \frac{1}{N} \sum_{i=0}^{N-1} |d_{\text{real}} - d_{\text{calc}}|, \quad \varepsilon \rightarrow \min, \quad (9)$$

where  $N$  is the number of objects;  $d_{\text{real}}$  is the true size of a given object;  $d_{\text{calc}}$  is the size of the object calculated using the homography matrix.

In this way, you can select the camera parameters using information about the sizes of objects shot. This algorithm allows you to calculate the transformation matrix for those cases when the configuration of the monocular system is unknown.

### *Detection and tracking*

To recognize an object in an image, algorithms based on machine learning are used. In order for the developed system to be implemented on low-yield computers, such as Jetson Orin Nano, the lightweight YOLOv8-nano architecture was chosen as a detector [16]. Unlike others, this architecture provides high-speed image processing in real time with sufficient accuracy [16, 17].

In this task, an experimental study of the algorithm was carried out in a warehouse, where forklifts with a typical pallet on forks were moving. The forklift was the object of interest, and the pallet was the object with known dimensions. Therefore, in this study, the YOLOv8-nano detector was trained to recognize the forklift in the image, and a segmenter of the same architecture and size was trained to search for the pallet.

About 15 000 images with a resolution of  $1920 \times 1080$  pixels were used to train the detector. The image data was acquired from different surveillance cameras installed in three warehouses. Before training, the input images were downscaled to  $640 \times 640$  pixels to speed up the detector. The training lasted 50 epochs and the data was split into training and validation sets in a ratio of 75 / 25. Fig. 3 shows an example of the data that was used to train the detector. The bounding boxes represent the output of the detector.

**Fig. 3.** Example data for training the detector.

To train the forklift pallet segmenter, about 30 000 images were cropped from the original resolution frames that were used to train the detector. The image data was pre-resized to  $640 \times 640$  pixels. The training lasted 30 epochs and the data was split into training and validation sets in a ratio of 75 / 25. Fig. 4 shows an example of the data that was used for training. The white area corresponds to the output of the segmenter.

**Fig. 4.** Example of data for training the segmenter.

To calculate the speed of an object, it is necessary to store the history of its positions for each moment of time when it is in the camera's field of view. To solve this problem, it is necessary to implement the tracker. In this study, we used the BoT-SORT technology [18], modified to solve the problem. The main modification is the introduction of a zone of interest, which is a polygon selected in the image. This zone is designed to filter detections of remote and difficult to distinguish objects. All objects recognized by the detector and falling into this zone are transmitted to the

tracker for further processing. Detections outside the zone of interest are not used in subsequent calculations. This modification allows us to avoid errors that may occur when processing objects that occupy too small an area in the image or are partially hidden behind the edge of the image. Also, our implementation of the tracker uses a Kalman filter implemented in C++ to speed up the work.

### *Selecting a key point of an object and refining the obtained coordinates on the image*

When selecting a key point for tracking, it is important to consider the perspective determined by the camera position. Possible distortions should be taken into account, and the points should be selected so as to minimize their influence when determining the location of the object of interest. The most reliable information about the position of a moving object on the observed plane is concentrated on the lower edge of the bounding box. Several key points can be identified on this segment: the middle, points associated with the object, and vertices.

The simplest solution is to use the middle of the lower edge as a reference point. However, this approach has a drawback: the position of this point on the observed plane relative to the object of interest will change depending on where the object is located in the frame.

Another option is to determine key points on the object located near or on the observed plane. This method is based on the fact that such points move together with the object, taking into account the camera perspective. An example of key points is the wheels of a car. The disadvantage of this method is that it is often difficult to find a point that is close to the observed plane and always visible to the camera.

The latest proposed solution is to use the far corner of the bounding box relative to the camera. Perspective distortions in this case have a minimal effect on determining the position of the object. To accurately determine the position of the far corner, you can take into account the camera position when calibrating it or use information about the trajectory of the object. For example, if the movement starts from the upper left corner and ends in the lower right, then the far corner will be the lower right corner of the frame.

To improve the accuracy of the obtained coordinates, an algorithm has been developed that reduces random errors that occur when using deep learning methods. The method is based on averaging the coordinate values based on a set of points. First, the size of the window for correction is set. Then this array is filled with points over several frames. As soon as a sufficient number of points have been accumulated, a linear regression is built on them, and the coordinates are corrected closer to this line, without going beyond the cumulative average error. The error value

is determined based on the error of the methods used to recognize objects in the image.

Fig. 5 shows the algorithm flow chart, where the rectangles represent the maximum correction for each axis, and the straight line is the linear regression constructed based on the coordinates in the image. The algorithm consists of the following steps:

- determining the window size for averaging;
- constructing a linear regression for all points in the window;
- correcting the coordinates based on the resulting regression.

**Fig. 5.** Correction of calculation points.

Circles — object detections on the image plane, rectangles — maximum permissible correction, straight line — linear regression

Using this averaging allows us to reduce the impact of the error of the methods used to recognize objects of interest in the image. A limitation of this approach is the requirement that the trajectory of the objects in the frame should be linear or change smoothly enough. As practice shows, this limitation is not strict. Often, the inertia of objects of interest, such as forklifts or cars, does not allow them to move along trajectories that are unsuitable for this algorithm.

### *Speed calculation*

After calculating the trajectory of an object on the observed plane and in the presence of the corresponding time marks, its average speed can be calculated. To do this, it is necessary to divide the movement of the object between two points by the time it took to cover this distance:

$$v_{\text{avg}} = \frac{|P_0 - P_1|}{\Delta t}, \quad (10)$$

where  $P_0, P_1$  are the coordinates of two points of the object's trajectory on the observed plane;  $\Delta t$  is the time it took the object to move from point  $P_0$  to point  $P_1$ .

Thus, to calculate the average speed of an object, it is necessary to store the history of its positions over time. An object tracker is used to solve this problem.

## RESULTS

### *Experimental conditions*

A series of experiments were conducted to test the performance of the algorithm and calculate the permissible limitations. The input data was the movement of forklifts with a pallet on forks in a warehouse. A series of video recordings of the drives were recorded. A configuration with the parameters specified in Tab. 1 was used as a monocular system.

**Tab. 1.** Monocular system parameters

The obtained video recordings were processed using the algorithm described above. About 2 h of real-time video were processed, which included 150 forklift passes in the camera's field of view. A zone of interest was preliminarily created for processing. This zone is shown in Fig. 6 and corresponds to the area bounded by the white polygon.

**Fig. 6.** Zone of Interest.

For all forklifts with a pallet on their forks that entered this area, the mask of the front part of the pallet was recognized. The mask was then employed to calculate the position of the forklift using the algorithm presented in the section "Algorithm description". In this case, the plane of the floor along which the forklifts moved served as the observed plane.

### *Calculation of the homography matrix*

In this experiment, the algorithm for automatic camera calibration was tested by iterative selection of parameters described in the section "Iterative selection of parameters for calculating the homography matrix". To conduct this test, we assume that the system parameters (except for the image resolution) are unknown in advance. The distribution of the calculated error depending on the camera parameters is shown in Fig. 7.

**Fig. 7.** Distribution of the calculated error during automatic calibration.

The absence of some values in this distribution (white area) is due to the fact that when using parameters from this area, the obtained distance to the object became less than or equal to the camera installation height, which is impossible.

In this case, the parameters that satisfy criterion (9) correspond with an error  $\varepsilon = 0.197$  to the following values:

$$\gamma = 65^\circ,$$

$$H = 6.8 \text{ м.}$$

These values are marked with a dot in Fig. 7. The star corresponds to the true parameters of the system, for them the value of the error function is  $\varepsilon = 0.297$ .

Such a difference between the calculated camera parameters and the true ones can be due to the similarity of the images depending on the camera position. Fig. 8 schematically depicts the described effect. With these configurations, the images will not differ much from each other. Such an effect is possible with small changes in the field of view angle  $\gamma$  and  $\gamma'$  when compared with the height  $H$  and  $H'$ . In other words, changing the field of view angle of the camera makes a much greater contribution to the final result of the calculated coordinates than changing the installation height. This statement is consistent with the parameters obtained during automatic calibration. The field of view angle differs by  $5^\circ$ , which is 8% from the true one, and the installation height by 3.8 m (227%).

**Fig. 8.** Difference in parameters of monocular system.

Next, we calculate the homography matrix for the true parameters specified in Tab. 1 and the calculated parameters of the monocular system. Fig. 9, a, shows the coordinate grid obtained as a result of transformation (2).

This grid displays perspective distortions introduced by the camera position. Each cell has dimensions of  $1 \times 1$  m. It is noticeable how the cell area decreases with increasing distance of the object from the camera along the observed plane. The grid displaying the resulting transformation matrix for the parameters obtained by the automatic calibration algorithm is shown in Fig. 9, б. This grid is more compressed compared to the one obtained using the true parameters. Next, we will consider the effect of the choice of calculated matrices when calculating the average speed of forklifts.

**Fig. 9.** Homography transformation.  
a — true parameters of the system,  
б — iterative selection algorithm

### Average speed calculation

#### Influence of homography matrix calculation

An experiment was conducted to calculate the average speed of passing forklifts in the camera's field of view. Homography matrices obtained using the true parameters of the monocular camera and the automatic calibration algorithm were applied for the calculation. In the experiment, the speed of the forklifts calculated using the proposed algorithm was compared with the speed obtained using GPS. The value obtained at the time the forklift left the area of interest was taken as the result of the algorithm.

Fig. 10 shows the distribution of speed calculation errors depending on the method used to construct the homography matrix. Solid columns correspond to the speeds obtained using the true camera parameters, dashed columns — using automatic calibration.

**Fig. 10.** GPS velocity distribution.

The vertex of the forklift's bounding box farthest from the camera was taken as key points. In this case, the obtained points were corrected in accordance with the algorithm described in the section "Selecting a key point of an object and refining the obtained coordinates on the image". It is noticeable that the dispersion of the values obtained using automatic calibration exceeds the values obtained using the true camera parameters. Tab. 2 shows the proportion of calculated velocities which error does not exceed a certain threshold.

**Tab. 2.** The influence of the homography matrix construction algorithm on the accuracy of the average speed calculation

When using the true parameters of the monocular system, 98% of the calculated velocities fall within the error limits of 1 km/h, and using the matrix obtained using automatic calibration allows us to achieve a metric of 84%. Therefore, using the automatic calibration algorithm significantly affects the

accuracy of speed calculation, but it can still be used in situations that do not require strong restrictions and when the system parameters are unknown.

#### Influence of correction of key points

Let us consider the influence of the key point correction algorithm on speed calculation. We will use the true system parameters to calculate the homography matrix. The lowermost corner of the bounding box will be used as a key point. Fig. 11 shows the result of calculating the average speed. The solid line corresponds to the calculation without the correction of key points, the dashed line — with correction, the dotted line — the speed according to GPS.

**Fig. 11.** Correction of calculation points when calculating speed.

In this case, the value of the average speed at each moment in time is calculated between the current position of the forklift in the frame and the initial position obtained at the moment of entering the zone of interest. The speed calculated using the correction algorithm turns out to be smoother compared to the speed calculated without correction.

Tab. 3 shows the proportion of calculated speeds, the error of which does not exceed a certain threshold, depending on the use of the correction algorithm of key points.

**Tab. 3.** Effect of correction of characteristic points on calculation of average speed

Using the correction algorithm, it is possible to increase the accuracy of calculating the average speed of forklifts. For example, for the threshold of 1 km/h, correction of points allows increasing the proportion of correct answers from 81% to 98%. As a result of point correction, it is possible to reduce the influence of random error caused by the algorithms for recognizing forklifts and the front part of the pallet in the image.

#### Influence of the selection of key points

Let us consider the influence of the selection of a characteristic point on the calculation of the average speed of forklifts in the area of interest. The matrix

obtained using the true parameters of the system was used as the homography matrix. In this case, the characteristic points were corrected in accordance with the algorithm described in the section "Selecting a key point of an object and refining the obtained coordinates on the image". The following were selected as key points: the center of the lower edge of the bounding box, the center of the front part of the pallet, and the far lower corner of the bounding box relative to the camera, in this case, the lower right vertex. Tab. 4 presents the proportion of calculated speeds with error that do not exceed a certain threshold, depending on the selected key point.

**Tab. 4.** Average speed deviation metric depending on characteristic points

As expected, the lowest accuracy was shown by the calculation for points located at the center of the lower edge of the bounding box. The highest number of calculated speeds with an error interval of 1 km/h was achieved when using the far corner of the frame and amounted to 98%. For weaker restrictions, the difference becomes smaller.

#### CONCLUSION

The paper proposes an algorithm for calculating the positions and speeds of objects on a plane based on data from a monocular camera. The algorithm is based on constructing a matrix for transforming points on the image plane into points on the observed plane along which objects move in the camera's field of view. The matrix is constructed using information on the real size of the object and the parameters of the monocular system — installation height and field of view angle. An algorithm for automatic camera calibration has also been developed. A neural network detector and segmenter were used as methods for recognizing objects in an image. To reduce the perspective error, a study was conducted to select a key point of an object in an image. To reduce the error of neural network recognition methods, an algorithm for correcting characteristic points based on averaging coordinate values was developed. An experiment was conducted to determine the average speed of forklifts moving in a distribution warehouse in the field of view of a monocular camera. The results of the experiment showed that 98% of the calculated average speeds were within  $\pm 1$  km/h of the reference values.