

Алгоритм обработки и преобразования кластерных радарных данных в объектовые данные с использованием математических и статистических методов

А.Д. Кузин^{1,2,3}, В.В. Дебелов^{1,2}, Д.В. Ендачёв^{1,2}

¹ *Московский политехнический университет, Москва, Россия;*

² *Центральный научно-исследовательский автомобильный и автомоторный институт «НАМИ», Москва, Россия;*

³ *Московский финансово-юридический университет, Москва, Россия*

АННОТАЦИЯ

Обоснование. В современных автономных транспортных системах, таких как беспилотные автомобили, важную роль играют радары, обеспечивающие обнаружение и классификацию объектов в окружающей среде. Однако данные с радаров часто содержат шумы и ошибки, что снижает точность детекции. Для повышения эффективности работы автономных систем требуется разработка алгоритмов, способных фильтровать и преобразовывать кластерные радарные данные в объектные данные, что позволит улучшить интерпретацию дорожной ситуации.

Цель работы — разработка алгоритма обработки данных радара, который обеспечит высокое качество результатов за счёт минимизации количества ошибок в детекции по сравнению с существующими подходами

Материалы и методы. В работе проведён анализ данных, полученных с автомобильного радара ARS 408 производства компании Continental Engineering Services, работающего на частоте 77 ГГц. Разработанный алгоритм включал этапы фильтрации по RCS, кластеризации и аппроксимации движения объектов. Для оценки эффективности алгоритма использовались метрики Precision, Recall и F1-score, а также анализ перекрытия IoU. Исследование проводилось на основе экспериментальных данных, собранных в условиях реального дорожного движения.

Результаты. Разработан алгоритм, позволяющий снизить ошибки обнаружения объектов. Оценка ошибок первого и второго рода показала, что предложенный метод обеспечивает более надёжное принятие решений автономными системами в различных дорожных условиях.

Заключение. Результаты позволяют сделать вывод о том, что разработанный алгоритм обработки радарных данных может быть успешно применён в системах автономного управления транспортными средствами, обеспечивая улучшение качества данных независимо от производителя радара. Практическая значимость заключается в возможности адаптации алгоритма к различным типам радаров, что делает его универсальным инструментом для повышения безопасности и эффективности автономных транспортных систем.

Ключевые слова: автономное вождение; системы детекции объектов; радарные данные; алгоритмы обработки данных; кластеризация; электротехнический комплекс.

Как цитировать:

Кузин А.Д., Дебелов В.В., Ендачёв Д.В. Алгоритм обработки и преобразования кластерных радарных данных в объектовые данные с использованием математических и статистических методов // Известия МГТУ «МАМИ». 2025. Т. 19, № 2. С. **x-y**. DOI: 10.17816/2074-0530-634761 EDN: RDDIXB

The Algorithm for Processing and Transforming Clustered Radar Data into Object Data Using Mathematical and Statistical Methods

Anton D. Kuzin^{1,2,3}, Vladimir V. Debelov^{1,2}, Denis V. Endachev^{1,2}

¹ *Moscow Polytechnic University, Moscow, Russia;*

² *Central Research Automobile and Automotive Engines Institute "NAMI", Moscow, Russia;*

³ *Moscow University of Finance and Law, Moscow, Russia*

ABSTRACT

BACKGROUND: In modern autonomous transportation systems, such as unmanned vehicles, radars are crucial in detecting and classifying objects in the surrounding environment. However, radar data often contain noise and errors, which reduces detection accuracy. To enhance the effectiveness of autonomous systems, development of the algorithms capable of filtering and transforming clustered radar data into object data is needed in order to improve the interpretation of road situations.

AIM: Development of the radar data processing algorithm that ensures high-quality results by minimizing the number of detection errors compared to existing approaches.

METHODS: The study involved an analysis of data obtained from the ARS 408 automotive radar manufactured by Continental Engineering Services, operating at a frequency of 77 GHz. The developed algorithm included stages of filtering based on RCS (Radar Cross Section), clustering, and objects motion approximation. To evaluate the algorithm's effectiveness, metrics such as Precision, Recall, and F1-score were used, along with the analysis of Intersection over Union (IoU). The research was conducted based on experimental data collected under real traffic conditions.

RESULTS: The work resulted in the development of the algorithm that reduces object detection errors. Evaluation of Type I and Type II errors demonstrated that the proposed method provides more reliable decision-making for autonomous systems in various road conditions.

CONCLUSION: The results support the conclusion that the developed radar data processing algorithm can be successfully implemented in autonomous vehicle control systems, providing improved data quality irrespective of the radar manufacturer. The practical significance lies in the ability to adapt the algorithm to various types of radar, making it a universal tool for enhancing the safety and efficiency of autonomous transportation systems.

Keywords: autonomous driving; object detection systems; radar data; data processing algorithms; clustering; electrotechnical facility.

TO CITE THIS ARTICLE:

Kuzin AD, Debelov VV, Endachev DV. The Algorithm for Processing and Transforming Clustered Radar Data into Object Data Using Mathematical and Statistical Methods. *Izvestiya MGTU «MAMI»*. 2025;19(2):x–y. DOI: [10.17816/2074-0530-634761](https://doi.org/10.17816/2074-0530-634761) EDN: RDDIXB

ВВЕДЕНИЕ

В современных автономных и беспилотных транспортных средствах сенсоры, ответственные за обнаружение объектов в окружающем пространстве, играют ключевую роль. Надёжность данных, получаемых от этих сенсоров, напрямую влияет на эффективность и безопасность работы систем. Однако данные с сенсоров часто содержат различные виды шумов, что снижает качество детекции объектов. Согласно классификации, разработанной Международной ассоциацией автомобильных инженеров (SAE International), автономные системы автомобилей классифицируются по пяти уровням автоматизации: от помощи водителю до полной автоматизации. Примеры таких систем включают адаптивный круиз-контроль, автоматическое торможение/смену полосы и парковочные автопилоты [2, 5, 11].

Одним из основных компонентов автономных систем на различных уровнях автоматизации является радар. Радар использует радиоволны для определения скорости и положения объектов, основываясь на их отражениях в пространстве и времени, за которое отражённые волны возвращаются обратно к радару. Однако эти радиоволны могут быть искажены различными факторами, такими как электромагнитные помехи, погодные условия или отражения от других объектов, что ведёт к возможной неверной интерпретации обстановки на дороге автономными системами автомобилей [3]. В радарных системах существует разнообразие методов обработки данных. Встроенные методы, предоставляемые производителями радаров, а также алгоритмы, используемые в системах беспилотного вождения для обработки полученных данных, хорошо известны [15]. Они демонстрируют относительно хорошую работоспособность, однако не всегда способны решить все задачи, связанные с обработкой радарных данных в условиях повышенного шума и сложных дорожных ситуаций [11].

В представленной работе решается задача повышения эффективности фильтрации данных путём уменьшения ошибок детекции на основе данных радара за счёт разработки алгоритма фильтрации данных с радара, работающего в кластерном режиме, включающего фильтрацию по эффективной площади рассеяния (RCS), кластеризацию данных и определение их положения. В данном контексте разрабатывается новый алгоритм, который направлен на решение указанных проблем и является более универсальным и адаптивным по сравнению с существующими методами [8].

Целевой задачей алгоритма фильтрации является обеспечение надёжной интерпретации данных радара, что позволяет системе автономного вождения принимать правильные решения в различных дорожных условиях. Алгоритм должен быть универсальным и применимым к любым типам радаров, выдающим данные о положении объектов, их скорости, сигнатуре и эффективной площади рассеяния (RCS) [4].

Актуальность проблемы состоит в том, что к "сырым" данным с радаров зачастую имеют доступ только сами производители радаров, что ограничивает возможности обработки этих данных напрямую. Существующие методы фильтрации готовых данных с радаров не всегда эффективны и основаны на нейронных сетях, что может быть ненадёжным решением. Таким образом, существует потребность в разработке алгоритма для уменьшения ошибок обнаружения и повышения качества данных, полученных с радара, независимо от производителя [6].

Целью данной работы является разработка универсального алгоритма обработки данных, поступающих от различных радиолокационных систем, представляющих информацию в виде облаков точек, заданных в двумерной декартовой системе координат, для обеспечения высокой сходимости при обнаружении объектов в условиях динамически изменяющейся окружающей среды и его применения в системах автономного управления транспортными средствами.

Разработка алгоритма обработки данных с радара

Разработка алгоритма по обработке радарных данных для фильтрации шумов проводилась на фрагменте лога данных, полученных с радара ARS 408, производства компании Continental Engineering Services. Данный радар обладает широким углом обзора и высокой чувствительностью, работая на частоте 77 ГГц. Согласно документации, радар ARS использует радиолокационное излучение для анализа окружающей среды [4, 12]. Полученные отражённые сигналы после нескольких этапов обработки алгоритмами самого радара становились доступными в виде кластеров и объектов.

Кластеры представляют собой отражённые радиолокационные волны с информацией о положении, скорости и уровне сигнала, оцениваемые заново каждый цикл. Объекты, в свою очередь, представляют собой группы кластеров, объединённые на основе пространственной близости и других характеристик [8]. В отличие от кластеров объекты имеют историю и размерность, что делает их предпочтительными для анализа объектов и дорожной ситуации.

Тем не менее, как и у любого датчика, у радара ARS 408 возникают проблемы с корректным обнаружением объектов, как показано на рис. 1.

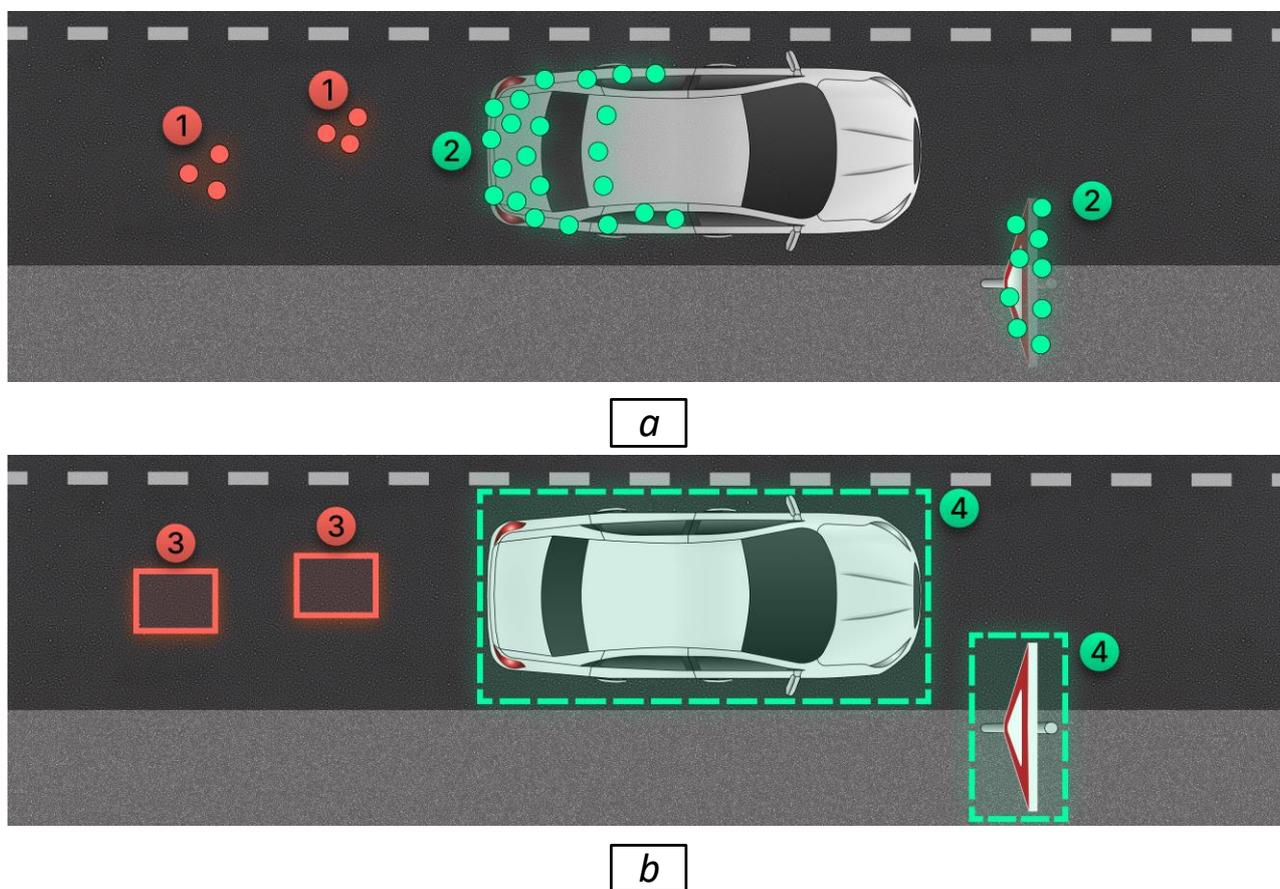


Рис. 1. Пример детекции объектов в кластерном и объектовом режиме: *a* — пример радарных данных в кластерном режиме работы; *b* — пример радарных данных в объектовом режиме работы: 1 — ложные объекты в кластерном режиме; 2 — истинные объекты в кластерном режиме; 3 — ложные объекты в объектном режиме; 4 — истинные объекты в объектном режиме. Истинные объекты — обнаруженные радаром объекты, соответствующие тем, которые пользователь ожидает или хочет зафиксировать; ложные объекты — регистрация радаром объектов, не представляющих интереса для пользователя, либо ошибочное детектирование шумов.

Fig. 1. Example of object detection by a vehicle radar in cluster and object modes: *a*: example of the radar data in the cluster mode; *b*: example of the radar data in the object mode: 1: false objects in the cluster mode; 2: true objects in the cluster mode; 3: false objects in the object mode; 4: true objects in the object mode. True objects: objects detected by the radar that correspond to those that the user expects or wishes to record; false objects: radar registered objects that are not of interest to the user or false detecting of noises.

Для решения данных проблем в радаре реализованы различные механизмы фильтрации сигналов. Табл. 1 содержит описание некоторых режимов фильтрации.

Таблица 1. Перечень режимов фильтрации в ARS 408

Table 1. List of filtration modes in the ARS 408

Критерий фильтрации	Кластер	Объект	Описание
NofObj	x	x	Ограничивает максимальное количество рассылаемых кластеров или объектов (минимальное значение игнорируется)
Расстояние	x	x	Радиальное расстояние в м [$r = \sqrt{x^2 + y^2}$]
Азимут	x	x	Угол в градусах [$\alpha = \text{arctg}(y/x)$]
VrelOncome	x	x	Радиальная скорость в зоне прямой видимости датчика в м/с приближающихся скоплений или объектов (все удаляющиеся скопления и объекты в порядке)
VrelDepart	x	x	Радиальная скорость в зоне прямой видимости датчика в м/с удаляющихся скоплений или объектов (все встречные скопления и объекты в порядке)
RCS	x	x	Значение RCS (поперечное сечение радара) в дБм ²
Lifetime		x	Время жизни (с момента первого обнаружения) в секундах
Size		x	Размер объекта как площадь в м ²
ProbExists		x	Вероятность существования, т. е. вероятность того, что это реальная цель, а не артефакт датчика

Наиболее значимыми способами фильтрации для разрабатываемого алгоритма являются фильтрация по RCS и по времени жизни объекта (параметр Lifetime). Фильтрация по RCS, согласно табл. 1, доступна как кластерам, так и объектам. Параметр Lifetime доступен только для объектов и означает, что объект считается существующим, если он подтверждён в течение установленного количества времени [12]. Оптимальное значение времени жизни объекта по документации ARS 408 составляет одну секунду, что минимизирует количество ложных объектов. Однако это время может быть критически важным в случае быстро движущихся или маневрирующих объектов, где даже небольшая задержка может иметь серьёзные последствия для безопасности. Кроме того, загрузка шины данных CAN, по которой происходит взаимодействие между радаром ARS 408 и системой управления автомобилем, также играет важную роль. Чем больше информации передаётся и получается с радара, тем меньше доступного пространства на шине данных, что также влияет на общую эффективность и безопасность системы автономного вождения.

Алгоритм обработки радарных данных

Алгоритм обработки данных разработан для повышения качества обнаружения объектов и оптимизации времени работы системы на основе данных радара ARS 408. Выбор кластерного режима обработки данных обоснован его способностью обеспечивать детализированное отслеживание каждой радиолокационной точки.

Разработанный алгоритм функционирует следующим образом: радар периодически передаёт данные в виде кластеров с интервалом около 75 миллисекунд, в соответствии с характеристиками, указанными в документации производителя. Каждая радиолокационная точка в кластере подвергается фильтрации на основе эквивалентной площади рассеивания (RCS), что позволяет выделить сигналы с высокой и низкой интенсивностью. Сигналы с высокой RCS формируют начальные кластеры. На каждом временном шаге новые кластеры объединяются с уже существующими с предыдущего шага. Кластер считается захваченным,

если его существование подтверждается в течение N последовательных обновлений данных радара. На следующем ($N+1$) обновлении данных для каждого захваченного кластера происходит поиск и добавление слабых сигналов, обнаруженных в его окрестности. Для захваченных кластеров вычисляются параметры направления движения, что позволяет окончательно идентифицировать объекты.

На рис. 2 представлена блок-схема разработанного алгоритма фильтрации кластерных данных с радара ARS 408 и преобразования их в объекты.

Формат данных

Для разработки алгоритма фильтрации радарных данных, функционирующего эффективно, необходимо детально описать данные, поступающие в кластерном режиме, которые используются в процессе обработки. Кластерный режим предоставляет информацию о точках, обнаруженных радаром, в виде кластеров, что позволяет проводить детальный анализ окружающей среды и объектов в ней.

Для тестирования алгоритма был получен набор данных в формате CSV, содержащий точки, полученные с радара за одно сканирование (скан). В табл. 2 представлен пример таких данных, полученных в кластерном режиме. Следует отметить, что датчик ARS408, согласно документации, способен фиксировать до 255 точек за один скан. Каждый скан поступает с радара с высокой частотой — примерно каждые 0,75 миллисекунды.

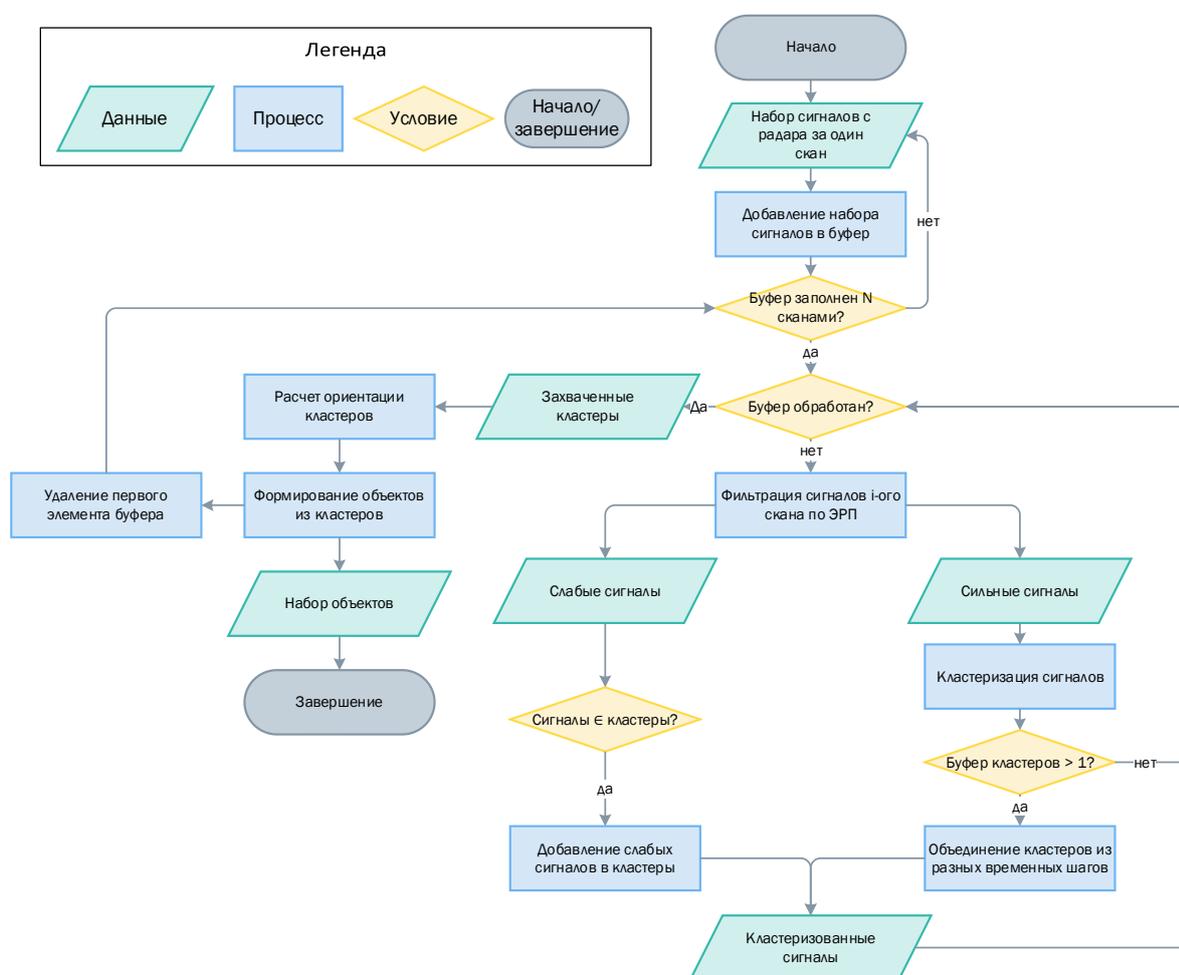


Рис. 2. Блок-схема алгоритма фильтрации кластерных данных с радара и их преобразования в объекты.

Fig. 2. Block diagram of the algorithm for filtering radar cluster data and transforming it into objects.

Таблица 2. Пример входных данных

Table 2. Example of input data

Cluster TimeStamp	Cluster DistLong	Cluster DistLat	Cluster VrelLong	Cluster VrelLat	Cluster RCS	Cluster DynProb	Time series	Velocity
7,02E+08	4,60001	-1,3	0	0	-9	1	1	23,33
7,02E+08	5,39999	-1,1	0	0	-16,5	1	1	23,33
7,02E+08	5	-3,1	0	0	-9,5	1	1	23,33
7,02E+08	7,20001	-1,5	0	0	7,5	1	1	23,33
7,02E+08	2,60001	-8,3	0	0	18	4	1	23,33
...
7,78E+08	46,2	-20,7	-8,5	1,25	0,5	3	260	54,21
7,78E+08	50,8	-7,1	-8,75	1,25	11,5	1	260	54,21
7,78E+08	48,6	-20,9	-8,25	1,25	1	3	260	54,21
7,78E+08	51,4	13,5	-9,5	1,5	5,5	1	260	54,21
7,78E+08	50,2	-20,7	-8,5	1,25	9	1	260	54,21

Каждая точка данных включает следующие параметры:

1. *TimeStamp* — временная метка сканирования радара в наносекундах;
2. *X* и *Y* — координаты объекта в метрах.
3. *V_x* и *V_y* — скорости объекта по осям *X* и *Y* в м/с.
4. *RCS* (Radar Cross Section) — радиолокационное сечение объекта, характеризующее его способность отражать радиосигналы.

5. *DynProp* — динамическое свойство объекта, указывающее на его текущее состояние и тип движения. Доступные значения включают:

- 0 (moving) — движущиеся объекты;
 - 1 (stationary) — неподвижные объекты;
 - 2 (oncoming) — объекты, движущиеся навстречу;
 - 3 (stationary candidate) — потенциально неподвижные объекты;
 - 4 (unknown) — объекты с неопределённым состоянием;
 - 5 (crossing stationary) — неподвижные объекты, находящиеся на пути движения;
 - 6 (crossing moving) — движущиеся объекты, находящиеся на пути движения;
 - 7 (stopped) — объекты, которые были в движении, но остановились.
6. *time series* — идентификатор временного шага, к которому относится объект.
 7. *Velocity* — скорость движения радара в м/с.

Описанные данные являются важной основой для анализа движения и классификации объектов в реальном времени. Использование динамического свойства значительно улучшает интерпретацию данных, позволяя более точно определять состояние и тип движения каждого объекта.

Пример логов, содержащих радарные данные, представленный на рис. 3, визуализирует радарные данные, а динамическое свойство объектов демонстрирует разнообразие состояний объектов по их сигнатуре.

Фильтрация данных по эффективной площади рассеяния

Для обеспечения качественной детекции объектов был разработан алгоритм фильтрации точек по значениям радиолокационного сечения (RCS) и динамическому свойству (*DynProp*). RCS является важным показателем отражающей способности объектов и измеряется в децибелах на квадратный метр (дБм²). Чем выше значение RCS, тем сильнее объект отражает радиосигналы и легче его обнаружить радаром.

Фильтрация данных по RCS осуществляется с использованием порогового метода, основанного на экспериментальных данных и анализе характеристик радара. Исследования показывают, что радар ARS 408 обладает высокой чувствительностью и может реагировать на различные объекты, включая капли дождя.

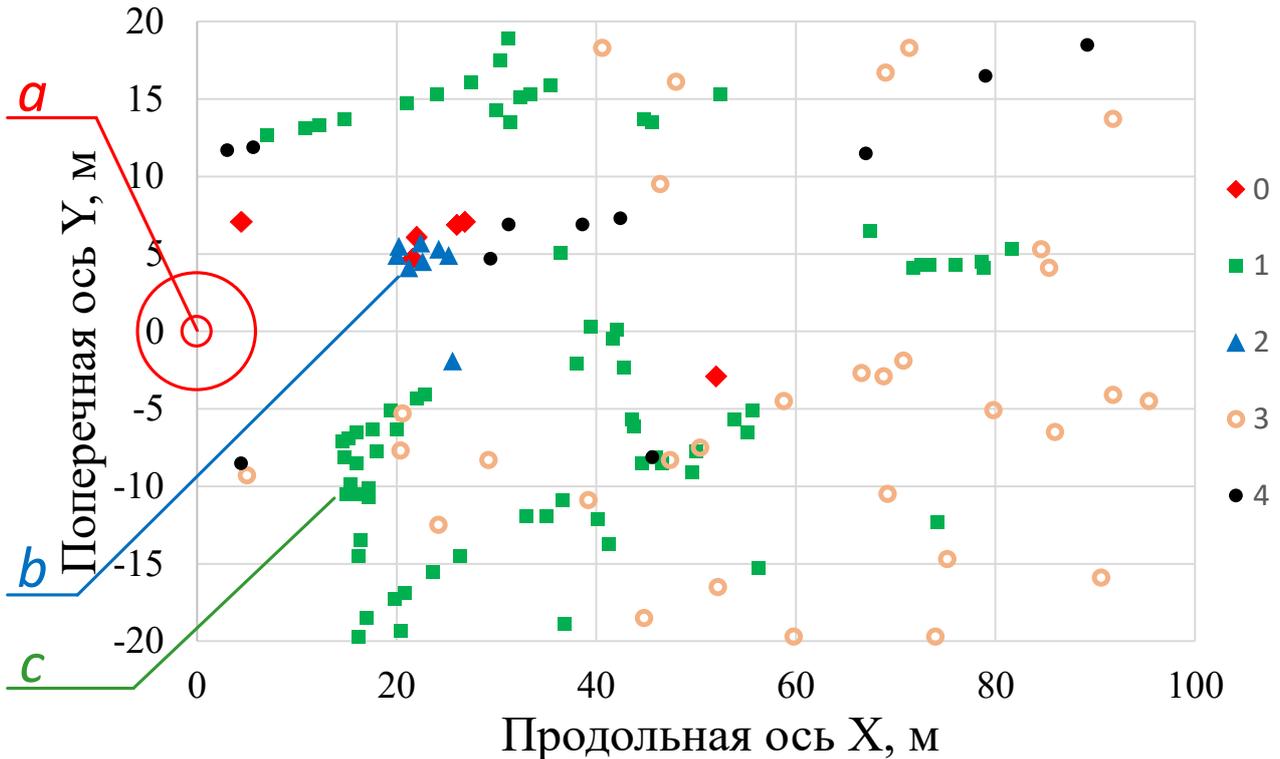


Рис. 3. Кластерные данные с радара за одно сканирование: *a* — местоположения радара относительно осей координат; *b* — динамические объекты, декретированные радаром; *c* — статические объекты, декретированные радаром: 0 — движущиеся объекты; 1 — неподвижные объекты; 2 — объекты, движущиеся навстречу; 3 — потенциально неподвижные объекты; 4 — неизвестный объект.

Fig. 3. Radar cluster data obtained during a single scan: *a*: radar position relative to the reference frame; *b*: dynamic objects detected by the radar; *c*: static objects detected by the radar: 0: moving; 1: stationary; 2: oncoming; 3: stationary candidate; 4: unknown.

Однако настолько незначительные объекты имеют низкое значение RCS, что делает их детекцию несущественно важным для автономного транспорта. Согласно документации ARS 408, динамические объекты имеют следующие характеристики, приведённые в табл. 3.

Таблица 3. Дальности обнаружения радиолокационного датчика ARS 408

Table 411. The detection ranges of the ARS 408 radar sensor

Угол	Автомобиль	Мотоцикл	Мопед	Пешеход
	10 дБ см (10 м ²)	5 дБ см (3 м ²)	0 дБ см (1 м ²)	-7 дБ см (0,2 м ²)
0°	170 м	150 м	110 м	75 м
±9°	140 м	110 м	80 м	55 м
±20°	70 м	60 м	45 м	30 м
±45°	55 м	40 м	30 м	20 м

Статические объекты, представляющие интерес для обработки, обычно имеют высокие значения RCS, так как они часто содержат металлические части. Поэтому объекты с RCS менее 10 дБм² обычно игнорируются в процессе обработки данных. Блок-схема алгоритма фильтрации представлена на рис. 4.

В процессе фильтрации основное внимание уделяется выявлению и отслеживанию динамических объектов, которые представляют наибольший интерес для автономных транспортных систем. Автомобили, мотоциклы и другие движущиеся объекты обычно имеют значения RCS в диапазоне от 1 до 10 дБм² в зависимости от их размера и характеристик.

Для определения порогового значения RCS для фильтрации данных использовались значения из табл. 1, отображающей максимальные расстояния обнаружения различных объектов при сканировании радаром под различными углами.

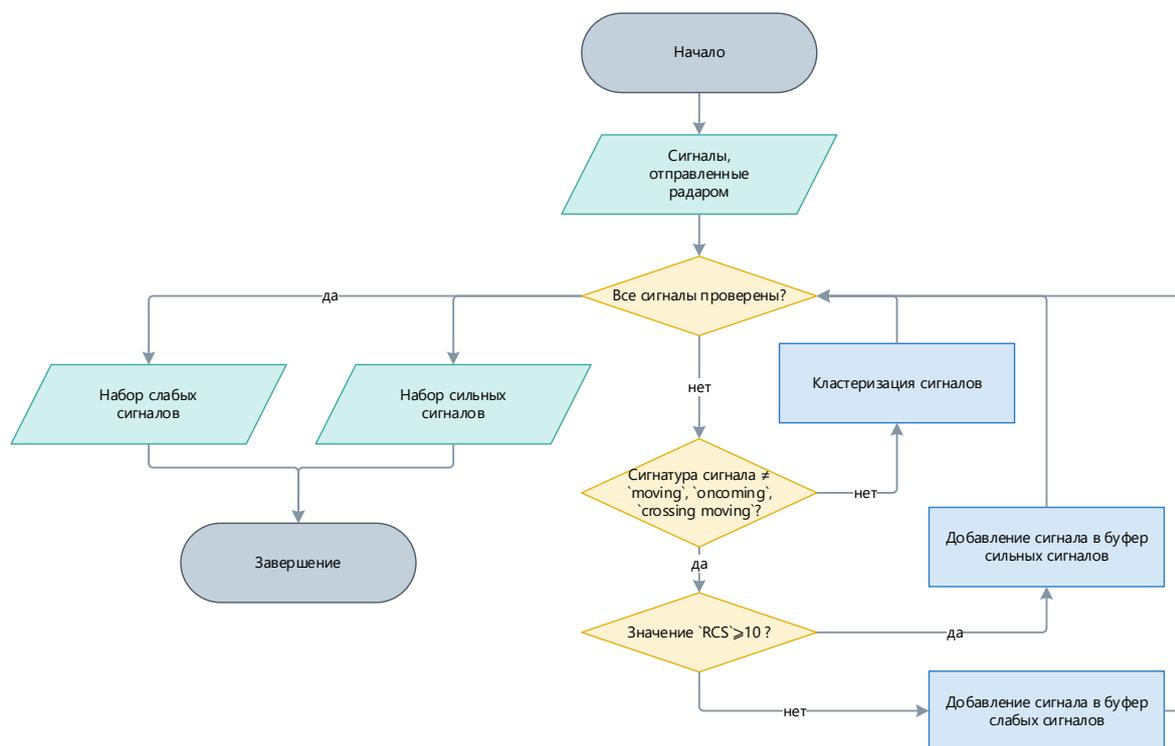


Рис. 4. Блок-схема алгоритма фильтрации по RCS.
Fig. 4. Block diagram of the RCS filtering algorithm.

Было установлено, что объекты с RCS менее 10 дБм², если они не движутся, не представляют интереса для детекции. Результаты фильтрации представлены на рис. 5.

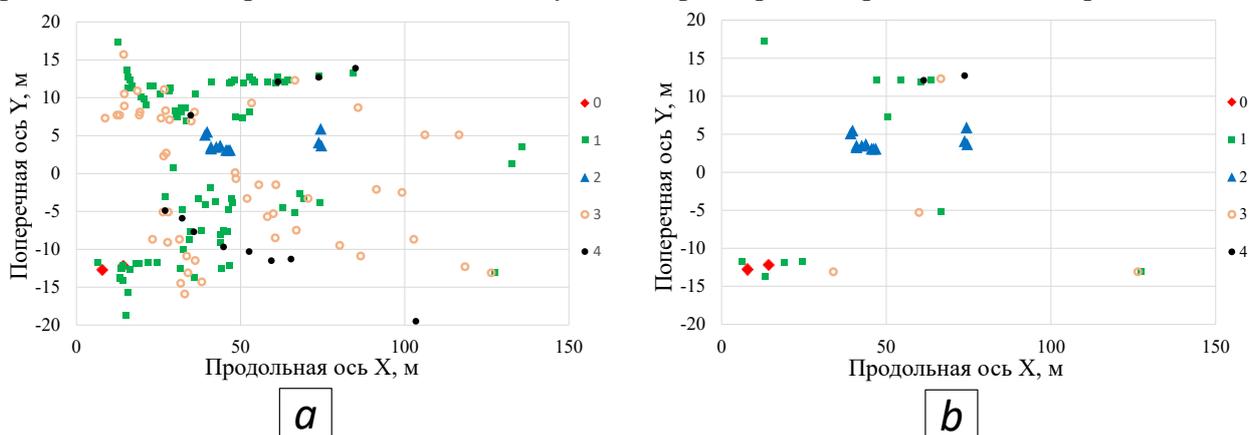


Рис. 5. Графики распределения сигналов на основе радарных данных до и после фильтрации по RCS: *a* — распределение сигналов с радарных данных до фильтрации по RCS; *b* — распределение сигналов с радарных данных после фильтрации по RCS: 0 — движущиеся объекты; 1 — неподвижные объекты; 2 — встречный объект; 3 — потенциально неподвижные объекты; 4 — неизвестный объект.

Fig. 15. Signal distribution plots based on the radar data before and after the RCS filtering: *a*: Signal distribution from the radar data before the RCS filtering; *b*: Signal distribution from the radar data after the RCS filtering: 0: moving; 1: stationary; 2: oncoming; 3: stationary candidate; 4: unknown.

Результаты анализа RCS и оценки динамического свойства интегрируются для принятия решения о том, какие цели следует отслеживать или обнаруживать, а какие могут быть проигнорированы.

Кластеризация данных и описание алгоритма

Анализ информации, получаемой от радарной системы, включает важный этап кластеризации, позволяющий выделить группы сигналов с похожими характеристиками и сформировать из них объекты [9]. Кластеризация помогает идентифицировать отдельные сигналы среди общего потока и упростить последующий анализ данных. Кластер представляет собой совокупность объектов, объединённых по схожим характеристикам, включая их пространственное расположение, скорость движения и сигнатуры сигналов.

Алгоритм кластеризации данных основывается на выявлении групп объектов с похожими характеристиками. Входными параметрами алгоритма являются множество точек со скана радара и пороговые значения для оценки схожести объектов [9]. На выходе алгоритм формирует группы объектов, объединённые на основе заданных пороговых значений.

Математическая формализация

Для алгоритма кластеризации потребуется следующий набор параметров.

1. Множество точек со скана радара $P = \{p_1, p_2, \dots, p_n\}$. Для алгоритма необходимо, чтобы каждая точка p_i содержала такие характеристики, как $\{x_i, y_i, v_i, s_i\}$, где x_i и y_i — координаты в пространстве, v_i — скорость, s_i — сигнатура сигнала.

2. Параметры ρ и ε — пороговые значения расстояния и разницы скоростей, используемые для оценки схожести объектов [9].

На выходе алгоритм формирует группы объектов, имеющих схожие характеристики, объединённые на основе заданных пороговых значений.

На выходе алгоритм предоставляет набор сформированных кластеров $C = \{c_1, c_2, \dots, c_M\}$, где каждый кластер c_k состоит из точек, имеющих схожие характеристики в пределах заданных пороговых значений.

Процесс кластеризации начинается с обработки набора сигналов (точек), полученных со скана радара. Алгоритм последовательно и систематически обходит все доступные сигналы с целью их распределения по соответствующим кластерам.

Для каждой точки (сигнала) из входного набора выполняется проверка: принадлежит ли она уже какому-либо сформированному кластеру?

– Если текущая точка уже отнесена к существующему кластеру, алгоритм пропускает её и переходит к следующему необработанному сигналу. Это предотвращает повторную обработку и дублирование точек в кластерах.

– Если точка ещё не кластеризована, она рассматривается как стартовая точка для формирования нового кластера. В этом случае:

1. Счётчик кластеров увеличивается, отмечая создание нового кластерного объекта.
2. Текущая точка добавляется в этот новый кластер.
3. Эта же точка помещается в очередь обхода — временное хранилище для сигналов, чьи непосредственные соседи будут проверяться на схожесть для расширения текущего кластера.

Далее алгоритм входит во внутренний цикл, предназначенный для расширения текущего кластера путём исследования соседних точек. Этот цикл продолжается до тех пор, пока очередь обхода не станет пустой:

– Пока очередь обхода не пуста, из неё извлекается одна точка. Эта точка будет использоваться для поиска своих соседей, которые могут быть включены в тот же кластер.

– Извлечённая точка сравнивается со всеми остальными некластеризованными сигналами в наборе данных на предмет удовлетворения условиям кластеризации. Эти условия

определяют, насколько сильно две точки схожи и могут ли они принадлежать одному кластеру. Схожесть оценивается на основе двух критериев:

1. Расстояние между двумя точками p_i и p_j определяется как

$$D(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

Если это расстояние не превышает заданный порог ρ , то точки считаются пространственно близкими.

2. Разница в скорости между двумя точками p_i и p_j вычисляется как

$$V(p_i, p_j) = |v_i - v_j| \quad (2)$$

Если эта разница не превышает порог ε , то точки считаются схожими по скорости.

– Если оба условия (по расстоянию и по разнице скоростей) выполнены для сравниваемой точки, то она считается схожей с извлечённой точкой и:

1. Добавляется в текущий формируемый кластер.
2. Также помещается в очередь обхода. Это ключевой шаг, который позволяет алгоритму "расширять" кластер, исследуя соседей вновь добавленных точек, обеспечивая нахождение всех связанных элементов.

– Если сравниваемая точка не удовлетворяет условиям кластеризации, она не добавляется в текущий кластер, и процесс поиска подходящих соседей для извлечённой точки продолжается.

Данный итеративный процесс продолжается до тех пор, пока очередь обхода не опустеет. Это означает, что для текущего кластера были найдены и обработаны все достижимые точки в пределах заданных критериев схожести. По завершении данного этапа алгоритм возвращается к обработке следующего необработанного сигнала из исходного набора данных, чтобы при необходимости инициировать формирование нового кластера.

Когда все сигналы просмотрены и распределены по соответствующим кластерам, алгоритм завершает свою работу, предоставляя итоговый набор сформированных кластеров объектов. Эти кластеры представляют собой группы точек, обладающих схожими характеристиками, которые были эффективно выделены из общего потока радарных данных.

Блок-схема алгоритма представлена на рис. 6.

Основным преимуществом разработанного алгоритма кластеризации является его способность учитывать, как пространственное, так и временное распределение объектов. Метод позволяет эффективно группировать объекты на основе их расположения и скорости движения.

Результат кластеризации представлен на рис. 7, отображающем объекты, ставшие более выраженными и организованными в группы. Визуально стало заметно, что объекты на изображении стали более выраженными и организованными в группы.

Алгоритм верификации устойчивости кластеров и описание алгоритма

Анализ данных в динамических системах, где объекты непрерывно перемещаются и изменяют свои свойства, представляет собой сложную задачу. Для эффективной обработки таких данных требуется разработка алгоритмов, способных выявлять устойчивые структуры и отфильтровывать нестабильные объекты [7, 9].

Алгоритм верификации устойчивости кластеров основан на принципе наблюдения и подтверждения постоянства объектов в течение нескольких временных интервалов [9].

Он включает следующие этапы.

1. Мониторинг кластера: кластер отслеживается по мере поступления данных на протяжении нескольких последовательных временных интервалов. На каждом интервале анализируется, продолжается ли существование кластера и соответствует ли он установленным критериям.

2. Подтверждение устойчивости: кластер считается подтверждённым, если он обнаруживается в данных на протяжении N последовательных интервалов. Значение N выбирается с учётом требуемого уровня надёжности и особенностей системы.

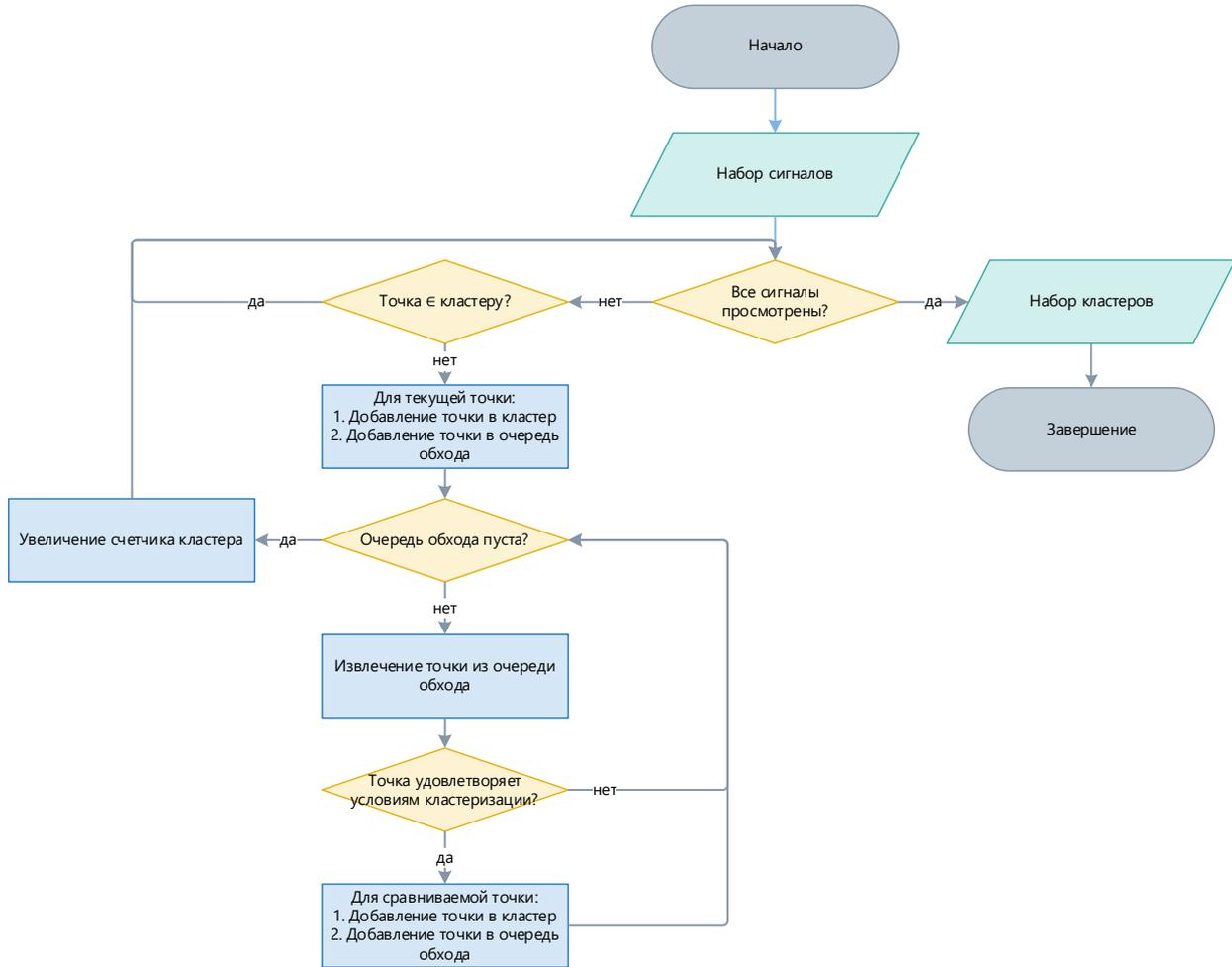


Рис. 6. Блок-схема алгоритма кластеризации.
Fig. 6. Block diagram of the clustering algorithm.

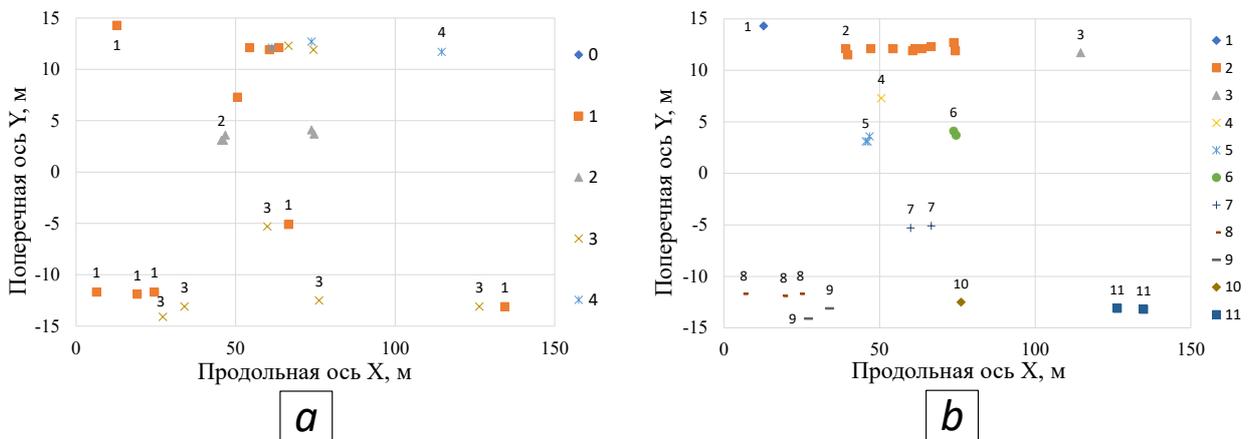


Рис. 7. Графики распределения сигналов на основе радарных данных до и после кластеризации: a — данные до кластеризации; b — данные после кластеризации.
Fig. 7. Signal distribution plots based on the radar data before and after clustering: a : Data before clustering; b : Data after clustering.

3. Поддержание существования: после подтверждения кластер сохраняется в системе, пока в его составе присутствует хотя бы одна точка. Если все точки кластера исчезают, он удаляется из системы.

4. Обновление состояния: каждый раз, когда новые данные поступают, состояние кластера обновляется.

На рис. 8 представлен алгоритм объединения.

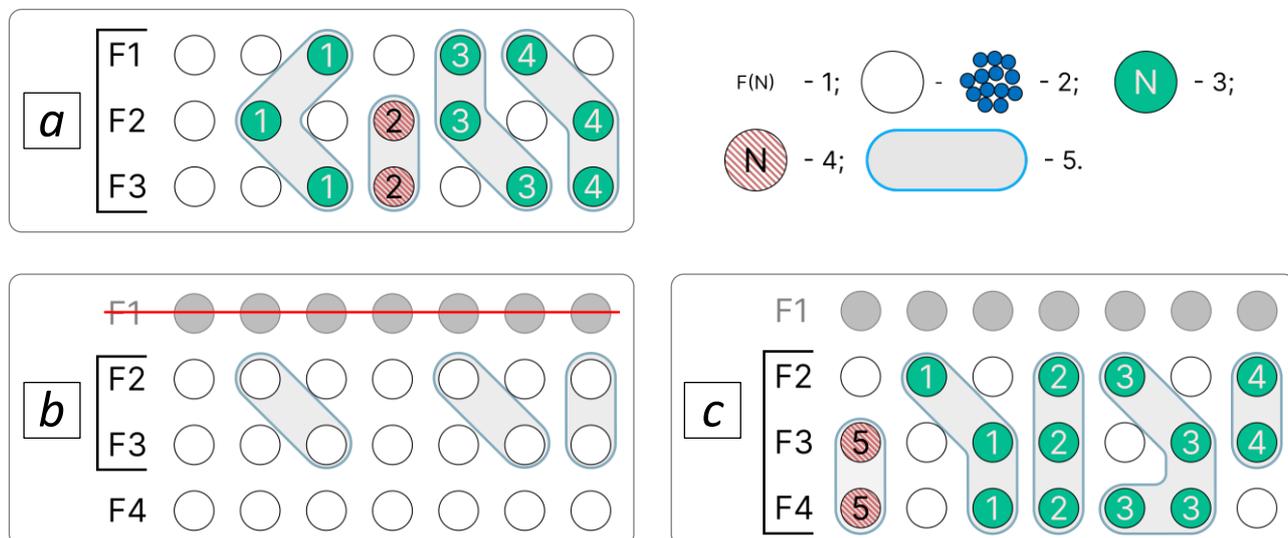


Рис. 8. Работа алгоритма объединения кластеризованных данных: *a* — первый цикл: накопилось 3 скана с разных временных шагов; *b* — удаление первого элемента из буфера и добавление нового; *c* — второй цикл: 1 — данные с радара, где *N* — номер скана; 2 — кластер, состоящий из точек; 3 — кластер подтвердился; 4 — кластер не подтвердился; 5 — кластеры объединились.

Fig. 8. Operation of the clustered data merging algorithm: *a*: First cycle: three scans accumulated from different time steps; *b*: Removal of the first element from the buffer and addition of a new one; *c*: Second cycle: 1: Radar data where *N* is the scan number; 2: Cluster consisting of points; 3: Cluster confirmed; 4: Cluster not confirmed; 5: Clusters merged.

Математическая формализация

Алгоритм объединения кластеров (как ключевой компонент верификации устойчивости) принимает на вход два набора кластеров: C_1 (например, кластеры с предыдущего временного шага) и C_2 (кластеры с текущего временного шага). Каждый кластер c_i из этих наборов состоит из набора точек $P = \{p_1, p_2, \dots, p_N\}$ и центра масс m_i . На выходе алгоритм формирует список объединённых кластеров, которые представляют собой совокупность объектов из предыдущего и текущего временных шагов.

Принцип работы алгоритма.

1. Сдвиг точек кластера (предсказание положения).

Для начала процесса объединения, для каждой точки p_i из кластеров C_1 (с предыдущего временного шага) выполняется предсказание её нового положения. Это делается путём "сдвига" точки в соответствии с её известной скоростью и разницей времени между текущим и предыдущим временными шагами.

– Расчёт Δt : время между последовательными сканами радара является важным параметром. Δt (в секундах) вычисляется как абсолютная разница между временными метками (*timeStamp*) текущего и предыдущего сканов, переведённая из наносекунд:

$$\Delta t = \frac{|timeStamp_n - timeStamp_{(n+1)}|}{(1 \cdot 10^9)} \quad (3)$$

где $timeStamp_n$ — время в наносекундах, в котором были получены данные со скана с радара; n — номер скана с радара.

– Вычисление новых координат: Новые (предсказанные) координаты точки $x_{predicted}$ вычисляются с учётом её текущего положения и скорости, умноженной на Δt :

$$x_{predicted} = x_i + v_i \cdot \Delta t \quad (4)$$

2. Сравнение точек (поиск соответствий):

– Для каждой точки p_k из кластера C_1 сравниваются все точки p'_j из кластера C_2 , рассчитывается расстояние между точками и скорость изменения расстояния по формулам 1 и 2.

3. Определение существования кластера.

Если найдены точки, расстояние и скорость изменения которых не превышают пороговые значения, то считается, что кластер существует на следующем временном шаге.

Объекты объединяются в новый кластер, представляющий собой совокупность объектов из предыдущего и текущего временных шагов.

Результат работы алгоритма и блок-схема алгоритма представлены на рис. 9 и 10.

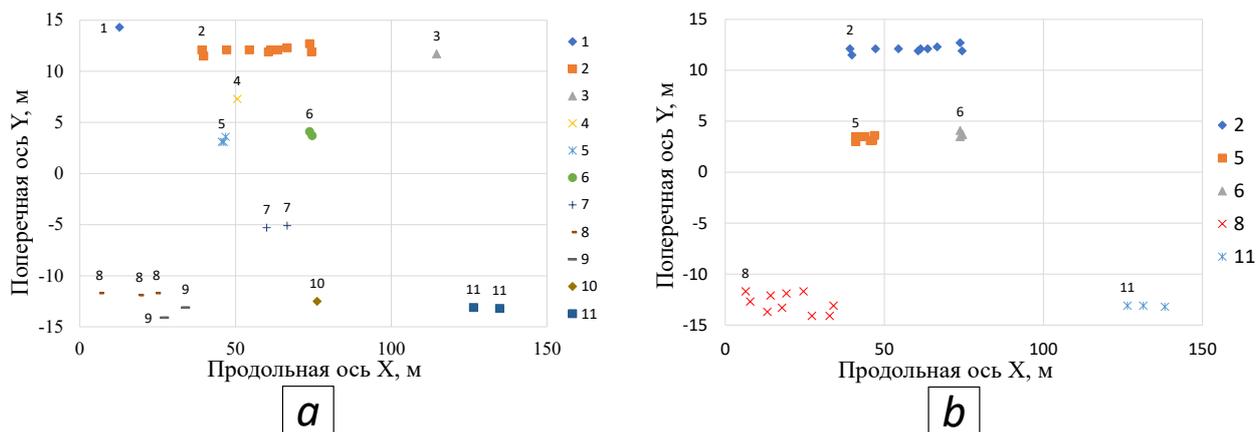


Рис. 9. Графики распределения сигналов на основе радарных данных до и после алгоритма с верификации устойчивости кластеров: *a* — данные до объектного трекинга; *b* — данные после объектного трекинга.

Fig. 9. Signal distribution plots based on the radar data before and after the cluster stability verification algorithm: *a*: Data before object tracking; *b*: Data after object tracking.

Алгоритм аппроксимации кластеров и описание алгоритма

Алгоритм аппроксимации кластеров аппроксимирует нейроны — движение кластера квадратичной функцией — и вычисляет ориентацию на основе аппроксимации. Он необходим для вычисления ориентации кластера в динамических системах, где объекты непрерывно перемещаются и изменяют свои свойства [1, 14].

Алгоритм аппроксимации кластеров основан на принципе нахождения квадратичной функции, которая лучше всего описывает движение кластера. Он включает следующие этапы.

1. Мониторинг кластера: кластер отслеживается по мере поступления данных на протяжении нескольких последовательных временных интервалов.

2. Аппроксимация кластера: движение кластера аппроксимируется квадратичной функцией, чтобы определить коэффициенты квадратичного члена, линейного члена и постоянного члена.

3. Вычисление ориентации: ориентация кластера вычисляется на основе аппроксимации, чтобы определить направление движения кластера.

На рис. 11 представлен алгоритм вычисления ориентации кластеров.

Математическая формализация

Алгоритм принимает на вход набор точек $P = \{p_1, p_2, \dots, p_N\}$, где каждая точка p_i состоит из координат X и Y , описывающих центр кластера. Набор точек описывает движение кластера.

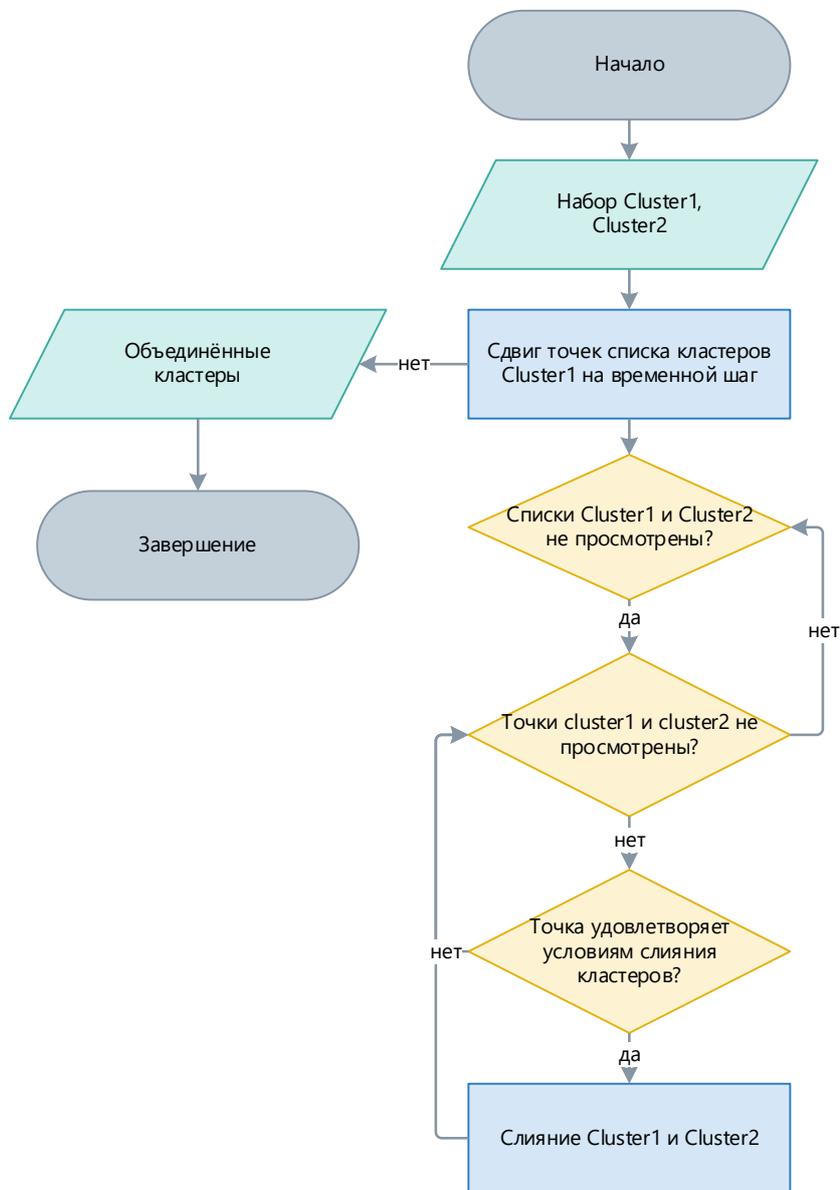


Рис. 10. Блок-схема алгоритма объединения кластеров.

Fig. 10. Block diagram of the cluster merging algorithm.

На выходе алгоритм формирует линейную или квадратичную функции, которые лучше всего описывают движение кластера, и вычисляет ориентацию кластера.

Принцип работы алгоритма.

1. Аппроксимация кластера: движение кластера аппроксимируется линейной функцией $y = ax + b$ и квадратичной функцией $y = ax^2 + bx + c$.

Для нахождения коэффициентов a , b и c используется метод наименьших квадратов: – для линейной аппроксимации:

$$\sum_{i=1}^n (y_i - ax_i - b)^2 \rightarrow \min, \quad (5)$$

$$a = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}, \quad (6)$$

$$b = \frac{\sum y_i - a \sum x_i}{n}; \quad (7)$$

– для квадратичной аппроксимации:

$$\sum_{i=1}^n (y_i - ax_i^2 - bx_i - c)^2 \rightarrow \min, \quad (8)$$

$$a = \frac{n \sum x_i^2 y_i - \sum x_i \sum y_i}{n \sum x_i^4 - (\sum x_i^2)^2}, \quad (9)$$

$$b = \frac{\sum x_i y_i - a \sum x_i^3}{\sum x_i^2}, \quad (10)$$

$$c = \frac{\sum y_i - a \sum x_i^2 - b \sum x_i}{n} \quad (11)$$

где n — количество точек; x_i и y_i — координаты i -й точки; a , b , c — коэффициенты квадратичной функции.

2. Выбор аппроксимации.

Для точного описания формы и движения кластеров, а также для определения их ориентации, необходимо выбрать подходящую математическую аппроксимацию. В данном алгоритме рассматриваются два основных типа аппроксимации: линейная и квадратичная. Выбор между ними производится на основе анализа соответствия аппроксимации данным и оценки ошибок.

2.1. Проверка положения минимума квадратичной функции

$$x_{\min} \in [\min(x), \max(x)] \wedge y_{\min} \in [\min(y), \max(y)], \quad (12)$$

$$x_{\min} = \frac{-b}{2a}, \quad (13)$$

$$y_{\min} = ax_{\min}^2 + bx_{\min} + c, \quad (14)$$

где x_{\min} — координата X точки минимума квадратичной функции; y_{\min} — координата Y точки минимума квадратичной функции; $\min(x)$, $\max(x)$ — минимальное и максимальное значения координаты X в наборе данных; $\min(y)$, $\max(y)$ — минимальное и максимальное значения координаты Y в наборе данных.

Данное условие используется для проверки, находится ли точка минимума квадратичной аппроксимации внутри диапазона данных. Если точка минимума квадратичной аппроксимации находится вне диапазона данных, то это означает, что квадратичная аппроксимация не может точно описать зависимости между данными в этом диапазоне. В этом случае выбирается линейная аппроксимация, которая является более простой и быстрой.

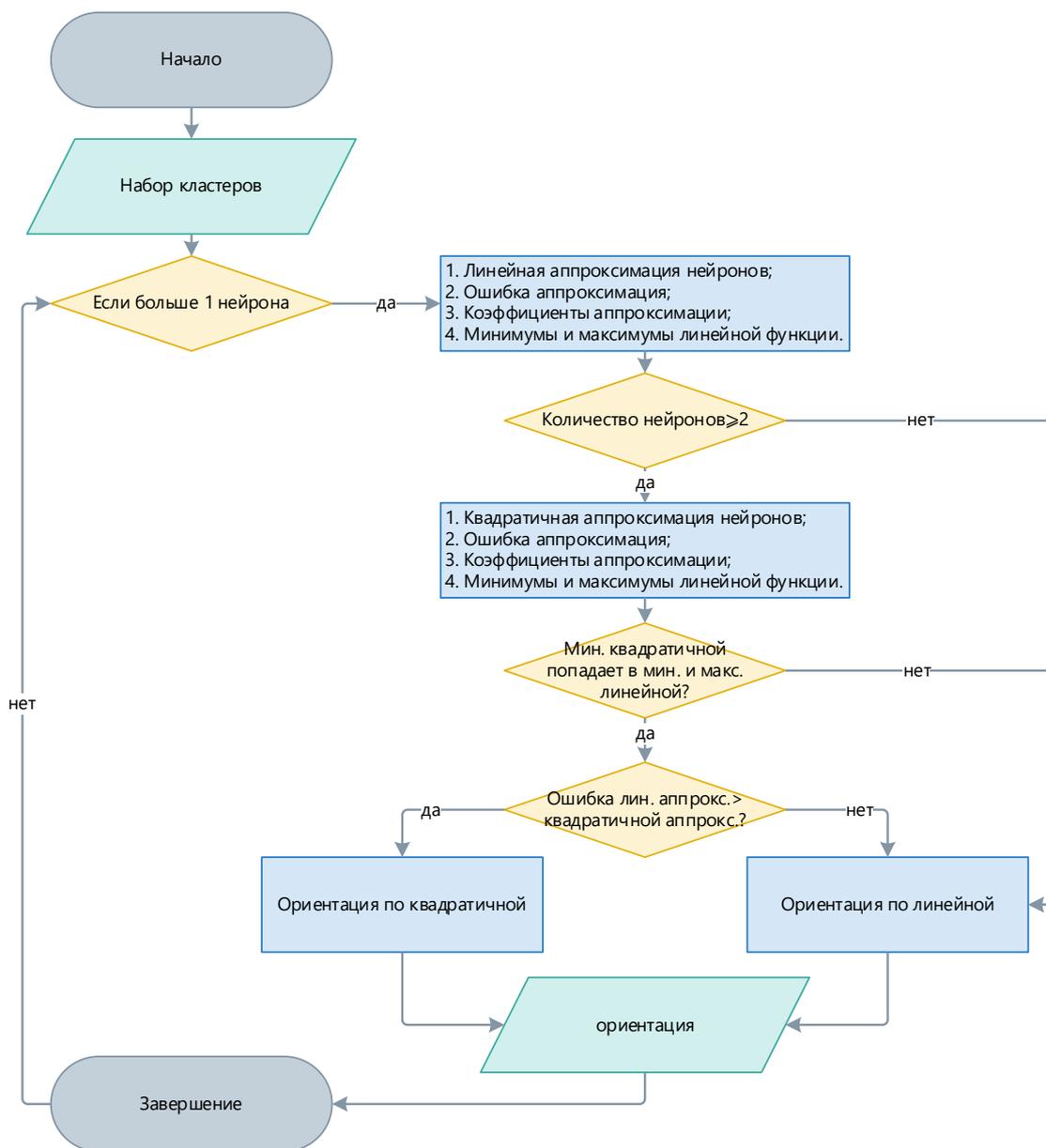


Рис. 11. Блок-схема вычисления ориентации кластеров.
Fig. 11. Block diagram for calculating clustering orientation.

2.2. Сравнение ошибок аппроксимаций

Если ошибка линейная больше ошибка квадратичная, то выбирается квадратичная аппроксимация. Иначе выбирается линейная аппроксимация.

$$f(x) = \begin{cases} ax^2 + bx + c, & \text{если } E_{\text{линейная}} < E_{\text{квадратичная}} \\ ax + b, & \text{иначе} \end{cases} \quad (15)$$

Для линейной аппроксимации:

$$E_{\text{линейная}} = \frac{1}{n} \sum_{i=1}^n (y_i - (ax_i + b))^2 \quad (16)$$

Для квадратичной аппроксимации:

$$E_{\text{квадратичная}} = \frac{1}{n} \sum_{i=1}^n (y_i - (ax_i^2 + bx_i + c))^2 \quad (17)$$

Линейная аппроксимация является более простой и быстрой, но она может не точно описывать сложные зависимости между данными. Квадратичная аппроксимация, с другой стороны, является более сложной и даёт больше информации, но она может быть более чувствительной к шуму в данных. Если ошибка линейной аппроксимации больше ошибки квадратичной аппроксимации, то это означает, что линейная аппроксимация не может надёжно описать зависимости между данными, и поэтому выбирается квадратичная аппроксимация.

3. Вычисление ориентации.

Ориентация кластера вычисляется на основе выбранной аппроксимации. Известно, где начало и конец цепочки точек кластера, и там мы рисуем касательную и вычисляем по ней угол.

Ориентация кластера вычисляется на основе выбранной аппроксимации.

Для линейной аппроксимации:

$$\varphi = \arctg(\alpha) \quad (18)$$

Для квадратичной аппроксимации:

$$\varphi = \arctg(ax_0 + b) \quad (19)$$

где φ — угол ориентации; α — наклон линейной функции; a и b — коэффициенты квадратичной функции; x_0 — координата x начальной точки цепочки точек.

На рис. 12 представлен иллюстративный пример результата функционирования алгоритма определения ориентации кластера.

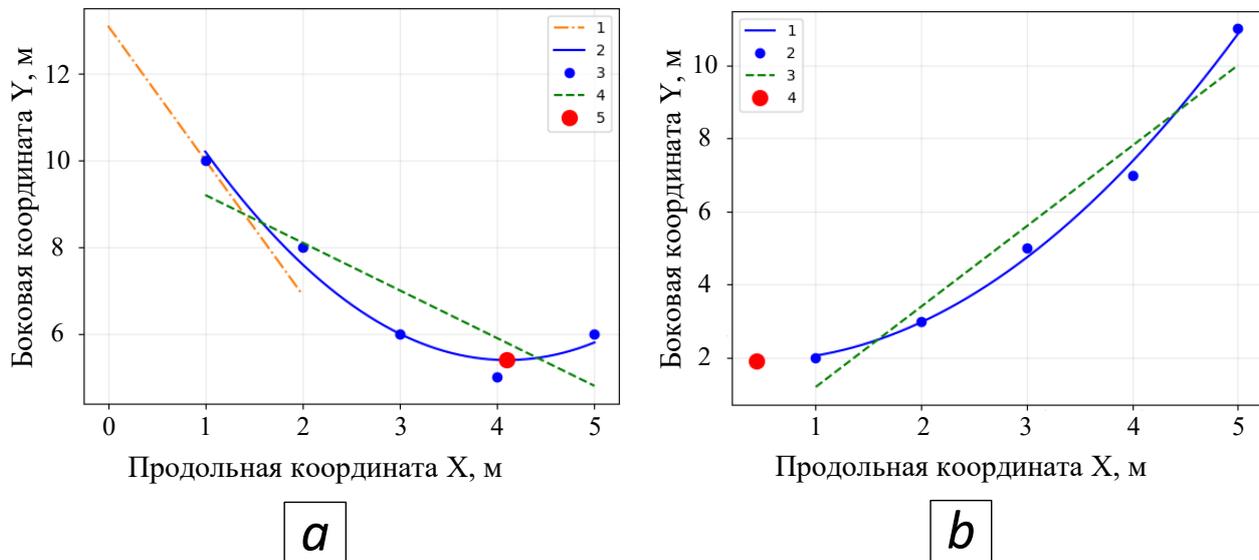


Рис. 12. Графики линейной и квадратичной аппроксимации нейронов кластера: *a*) кластер с нелинейной траекторией: 1 — касательная к квадратичной аппроксимации, определяющая основное направление движения кластера; 2 — квадратичная аппроксимация движения кластера; 3 — радарные точки аппроксимируемого кластера; 4 — линейная аппроксимация, используемая для сравнения или в качестве запасного варианта определения направления; 5 — точка экстремума квадратичной функции; *b*) кластер с преимущественно прямолинейной траекторией: 1 — касательная к линейной аппроксимации, определяющая ориентацию кластера; 2 — радарные точки аппроксимируемого кластера; 3 — выбранная линейная аппроксимация движения кластера; 4 — точка экстремума квадратичной функции.

Fig. 12. Plots of linear and quadratic approximation of cluster neurons: *a*: cluster with a non-linear trajectory: 1: tangent to the quadratic approximation, defining the cluster's primary direction of motion; 2: quadratic approximation of cluster motion; 3: radar points of the approximated cluster; 4: linear approximation, used for comparison or as a fallback for direction determination; 5: extremum point of the

quadratic function; b : cluster with a predominantly linear trajectory; l : tangent to the linear approximation, defining the cluster's orientation; 2: radar points of the approximated cluster; 3: selected linear approximation of cluster motion; 4: extremum point of the quadratic function.

Преобразование данных в объекты

Для дальнейшей работы беспилотной системы Autoware с данными нужно преобразовать их в объекты. В этом случае используется алгоритм преобразования кластеров в объекты. В процессе преобразования вычисляются параметры объекта на основе кластера точек. Скорость объекта позволяет классифицировать объект как динамический или статический.

Алгоритм принимает на вход кластер $C = \{c_1, c_2, \dots, c_N\}$, где c_i — точка кластера, а также $velocity$ — скорость в м/с и θ — угол в градусах.

На выходе алгоритм формирует объект $object = \{x, y, w, h, \theta, velocity\}$, где w и h — длина и ширина объекта.

Сначала алгоритм вычисляет центр кластера, используя средние значения координат x и y всех точек в кластере. Затем, если скорость объекта больше единицы, определяются ширина и высота объекта при помощи вычисления разницы между максимальными и минимальными значениями координат x и y соответственно. Если скорость объекта меньше единицы (т. е. объект статический), алгоритм использует алгоритм Грэхема для вычисления точек, составляющих выпуклую оболочку кластера точек, что позволяет определить форму объекта на основе его контурных точек [13].

На рис. 13 представлен иллюстративный пример преобразования кластеров в объекты.

Анализ результатов работы алгоритма

Результаты работы алгоритма фильтрации шумов радарных данных подверглись обработке с целью оценки его эффективности в обнаружении объектов. Для этого были произведены расчёты метрик Precision, Recall и F1-score [10].

– Precision: доля верно обнаруженных объектов среди всех объектов, обнаруженных алгоритмом.

– Recall: доля действительно существующих объектов, обнаруженных алгоритмом, среди всех эталонных объектов.

– F1-score: гармоническое среднее между Precision и Recall, позволяющее оценить баланс между ними.

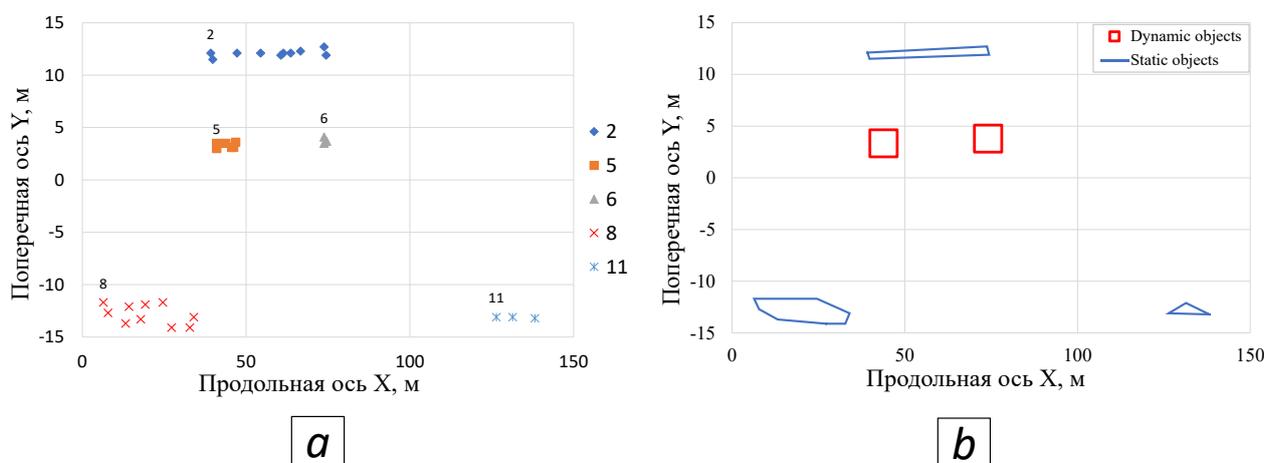


Рис. 13. Результат преобразования кластеров в объекты: a — данные после объектного трекинга; b — данные после формирования объектов.

Fig. 13. Results of converting clusters into objects: a : Data after object tracking; b : Data after object formation.

Анализ проводился на основе сопоставления результатов алгоритма с эталонными данными, а также путём расчёта меры перекрытия IoU (Intersection over Union) для оценки степени пересечения объектов на изображениях. Результаты работы алгоритма представлены на рис. 14 и в табл. 4.

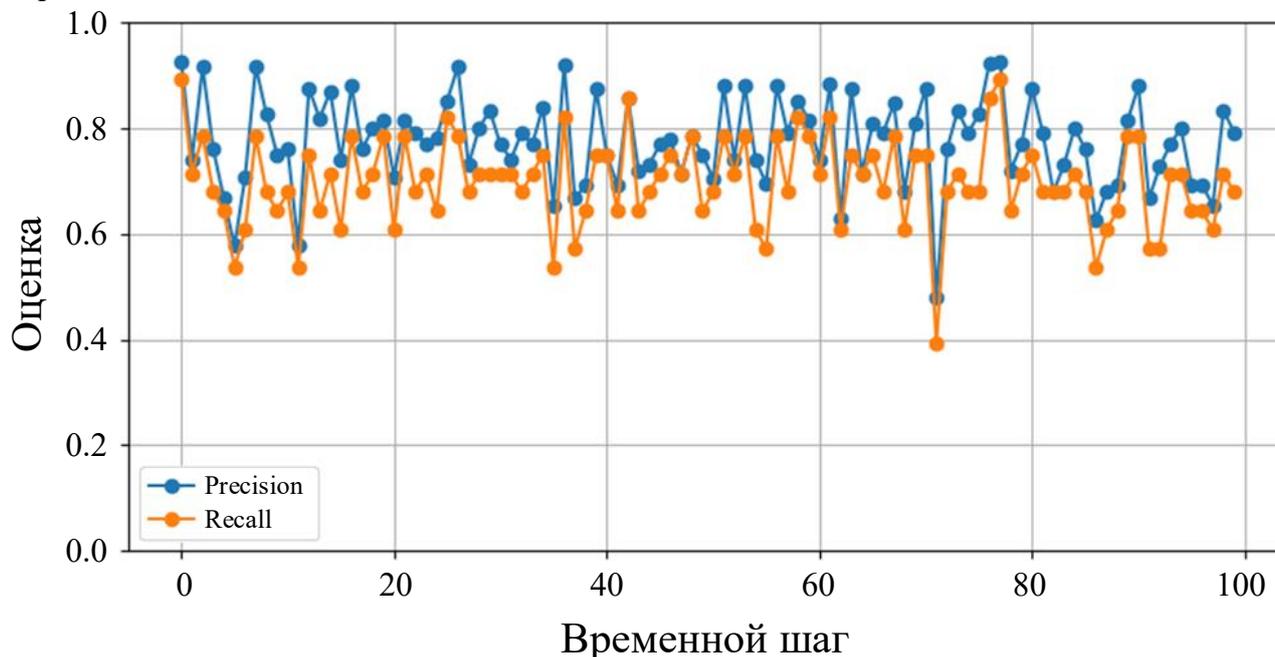


Рис. 14. График зависимости Precision и Recall от временного шага: Precision — доля верно обнаруженных объектов среди всех объектов, обнаруженных алгоритмом, что указывает на точность положительных предсказаний; Recall — доля действительно существующих объектов, обнаруженных алгоритмом, среди всех эталонных объектов.

Fig. 14. Plot of Precision and Recall evaluation metrics versus time step: Precision: the ratio of true positive predictions to the total predicted positives, indicating the accuracy of positive predictions; Recall: the ratio of true positive predictions to the total actual positives.

Таблица 4. Матрица ошибок для разработанного алгоритма

Table 4. Error matrix for the developed algorithm

Статус	Объект обнаружен	Объект не обнаружен
Объект существует	TP=2201	FP=899
Объект не существует	FN=591	TN=-

Значения метрик:

– Precision: 0.79;

– Recall: 0.71;

– F1-score: 0.75.

Разработанный алгоритм демонстрирует удовлетворительную эффективность в обнаружении объектов на радарных данных. Результаты анализа, представленные в виде метрик Precision, Recall и F1-score, показывают, что алгоритм верно обнаруживает 79% объектов из всех, найденных им, и 71% действительно существующих объектов. Гармоническое среднее этих метрик (F1-score) составляет 0,75, что свидетельствует о сбалансированной работе алгоритма.

ЗАКЛЮЧЕНИЕ

В рамках данного исследования был разработан и реализован алгоритм обработки данных с радара. Основной целью алгоритма являлась корректная детекция объектов на основе кластерных данных. Выбор кластерного режима радара для обработки данных

обусловлен его преимуществами перед объектным режимом, так как кластерный режим позволяет более гибко работать с данными и рассматривать каждый полученный сигнал как отдельный объект.

Алгоритм обработки данных включает несколько ключевых этапов: фильтрацию данных по эффективной площади рассеяния (RCS) и динамическим свойствам объектов, кластеризацию данных для выделения схожих групп объектов и преобразование кластеров в объекты. Результаты работы алгоритма были проанализированы с использованием метрик Precision, Recall и F1-score, которые показали высокую производительность алгоритма в детекции объектов на основе радарных данных. Сравнение текущих результатов с предыдущими данными продемонстрировало значительное улучшение производительности алгоритма.

Разработанный алгоритм представляет собой эффективный инструмент для обнаружения объектов и может быть успешно применён в системах автономного управления и беспилотных транспортных средствах. Полученные результаты служат основой для дальнейшего совершенствования и оптимизации алгоритма с целью повышения точности детекции объектов.

Перспективы применения данной работы включают системы автономного управления автомобилями и беспилотными транспортными средствами, автоматизированные системы мониторинга и управления дорожным движением, роботизированные системы, военные и оборонные приложения, а также системы безопасности и наблюдения, использующие радары для детекции и классификации движущихся объектов.

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Вклад авторов. Д.В. Ендачев — формулирование целей, задач исследования, руководство научным проектом, участие в обсуждении материалов статьи; В.В. Дебелов — научное руководство, формулирование основных направлений исследования, планирование исследований, редактирование статьи, участие в обсуждении материалов статьи; А.Д. Кузин — проведение исследований и экспериментов, обработка результатов исследования, подготовка и написание текста статьи. Все авторы одобрили рукопись (версию для публикации), а также согласились нести ответственность за все аспекты работы, гарантируя надлежащее рассмотрение и решение вопросов, связанных с точностью и добросовестностью любой её части.

Этическая экспертиза. Неприменимо.

Источники финансирования. Исследование и подготовка публикации проведены в рамках опытно-конструкторских работ и поисковых исследований в рамках НТЗ.

Раскрытие интересов. Авторы заявляют об отсутствии отношений, деятельности и интересов за последние три года, связанных с третьими лицами (коммерческими и некоммерческими), интересы которых могут быть затронуты содержанием статьи.

Оригинальность. При создании настоящей работы авторы не использовали ранее опубликованные сведения (текст, иллюстрации, данные).

Доступ к данным. Редакционная политика в отношении совместного использования данных к настоящей работе не применима, новые данные не собирали и не создавали.

Генеративный искусственный интеллект. При создании настоящей статьи технологии генеративного искусственного интеллекта не использовали.

Рассмотрение и рецензирование. Настоящая работа подана в журнал в инициативном порядке и рассмотрена по обычной процедуре. В рецензировании участвовали два внешних рецензента, член редакционной коллегии и научный редактор издания.

ADDITIONAL INFORMATION

Author contributions: D.V. Endachev: formulation of research goals, objectives, management of a scientific project, participation in the discussion of the materials of the manuscript; V.V. Debelov: scientific guidance, formulation of the main directions of the research, research planning, editing of the article, participation in the discussion of the materials of the manuscript; A.D. Kuzin: conducting research and experiments, processing

research results, preparation and writing of the text of the manuscript. All the authors approved the version of the manuscript to be published and agreed to be accountable for all aspects of the work, ensuring that issues related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Ethics approval: N/A.

Funding sources: The research and preparation of the publication were carried out within the framework of experimental design and exploratory research within the framework of the technological advance.

Disclosure of interests: The authors have no relationships, activities, or interests for the last three years related to for-profit or not-for-profit third parties whose interests may be affected by the content of the article.

Statement of originality: No previously obtained or published material (text, images, or data) was used in this study or article.

Data availability statement: The editorial policy regarding data sharing does not apply to this work as no new data was collected or created.

Generative AI: No generative artificial intelligence technologies were used to prepare this article.

Provenance and peer review: This paper was submitted unsolicited and reviewed following the standard procedure. The peer review involved two external reviewers, a member of the editorial board, and the in-house scientific editor.

СПИСОК ЛИТЕРАТУРЫ | REFERENCES

1. Vertkova MS. Linear and Quadratic Approximations. *Proceedings of the 63rd Student Scientific and Practical Conference of the Engineering Faculty of Samara State Agricultural Academy*. 2018:244–248. (In Russ.) EDN: VMLGYK
2. Debelov VV, Ivanov VV, Kozlovsky VN, et al. Modeling of an Electronic Speed Control System of a Passenger Car in Speed Maintenance and Limitation Modes. *Electronics and Electrical Equipment of Transport*. 2013;(6):2–7. (In Russ.) EDN: RTPZML
3. Debelov VV, Endachev DV, Evgrafov VV, et al. *Implementation Ways of Control Systems for Vehicle Motion Modes Based on Autonomous Driving Technology Development*. Safety of Wheeled Vehicles under Operating Conditions. 2017. P. 504–512. (In Russ.) EDN: ZFNVAZ
4. Debelov VV, Ivanov VV, Kozlovsky VN, et al. Electronic Speed Control System of a Vehicle in Speed Maintenance and Limitation Modes. *Truck*. 2013;(12):19–23. (In Russ.) EDN: RUINER
5. Deev OI, Debelov VV, Endachev DV. *Development of a Machine Vision System Using the Principle of Analysis of Changing Geometry and Spectrum of Incident Light*. Intelligent Transport Systems. 2017. P. 239–256. (In Russ.) EDN: YUFRTB
6. Endachyov DV, Bakhmutov SV, Evgrafov VV, et al. Electronic Systems of Intelligent Transport Vehicles. *Mechanics of Machines, Mechanisms and Materials*. 2020;(4):5–10. doi: 10.46864/1995-0470-2020-4-53-5-10 (In Russ.) EDN: IGEFKA
7. Ershov KS, Romanova TN. Analysis and Classification of Clustering Algorithms. *New Information Technologies in Automated Systems*. 2016;(19):274–279. (In Russ.) EDN: VTZOPF
8. Kozlovsky VN, Debelov VV, Deev OI, et al. Prospective Diagnostic Systems for Autonomous Transport Object Control. *Truck*. 2017;(6):21–28. (In Russ.) EDN: ZQPXKZ
9. Kuznetsov DYU, Troshina TL. Cluster Analysis and Its Application. *Yaroslavl Pedagogical Bulletin*. 2006;(4):103–107. (In Russ.) EDN: HWDSNJ
10. Kuznetsova AA. Statistical Precision-Recall Curves for Object Detection Quality Analysis. *Applied Informatics*. 2020;15(6):42–57. doi: 10.37791/2687-0649-2020-15-6-42-57 (In Russ.) EDN: VEPVNN
11. Saikin AM, Endachyov DV, Evgrafov VV, et al. Fundamentals of Evaluation Criteria for Unmanned Vehicles. *Proceedings of NAMI*. 2017;(2):48–54. (In Russ.) EDN: ZCSPAJ
12. Standardized Interface ARS Technical Documentation [Electronic resource] // GitHub. 2017. URL: https://github.com/lf2653/myrepository/blob/master/documentation/Short-Description_2017_07_09_06_ARS404-21%2BARS408-21_en_V1.06.pdf (accessed 28.07.2024). (In Russ.)
13. Chadnov RV, Skvortsov AV, Mirza NS. Review of Algorithms for Constructing Convex Hulls on the Plane. *Bulletin of Tomsk State University*. 2004(S9-2):116–121. (In Russ.) EDN: ZJFGSB
14. Benigno P, Woodford M. Linear-Quadratic Approximation of Optimal Policy Problems. *Journal of Economic Theory*. 2012;147(1):1–42. doi: 10.3386/w12672
15. Deev OI, Debelov VV, Endachev DV, et al. Technological Vision Systems Using Principle of Analysis of Changing Geometry and Spectrum of the Light. *IOP Conference Series: Earth and Environmental Science*. 2021;666(3):032075. doi: 10.1088/1755-1315/666/3/032075 EDN: MPQTJH

ОБ АВТОРАХ / AUTHORS' INFO

*** Кузин Антон Дмитриевич,**
инженер Центра электронных устройств;
адрес: Россия, 125438, Москва, ул.
Автомоторная, д. 2;
ORCID: 0009-0005-3342-8526;
eLibrary SPIN: 6493-7201;
e-mail: anton.kuzin@nami.ru

Соавторы:

Дебелов Владимир Валентинович,
начальник отдела технологии программного
обеспечения центра программного
обеспечения;

ORCID: 0000-0001-6050-0419;

eLibrary SPIN: 8701-7410;

e-mail: vladimir.debelov@nami.ru

Ендачѐв Денис Владимирович,

исполнительный директор по
информационным и интеллектуальным
системам;

ORCID: 0000-0003-3547-7928;

eLibrary SPIN: 6514-7752;

e-mail: denis.endachev@nami.ru

* Автор, ответственный за переписку / Corresponding author

*** Anton D. Kuzin,**
Engineer of the Electronic Devices Center;
address: 2 Avtomotornaya st, Moscow, Russia,
125438;

ORCID: 0009-0005-3342-8526;

eLibrary SPIN: 6493-7201;

e-mail: anton.kuzin@nami.ru

Co-Authors:

Vladimir V. Debelov,
Head of the Software Technology Department of
the Software Center;

ORCID: 0000-0001-6050-0419;

eLibrary SPIN: 8701-7410;

e-mail: vladimir.debelov@nami.ru

Denis V. Endachev,

Executive Director for Information and Intelligent
Systems;

ORCID: 0000-0003-3547-7928;

eLibrary SPIN: 6514-7752;

e-mail: denis.endachev@nami.ru