

ЭЛЕКТРОННАЯ КОМПОНЕНТНАЯ БАЗА
И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

УДК 004.896

СРАВНЕНИЕ МЕТОДОВ СЕГМЕНТАЦИИ ПОВЕРХНОСТИ
С ДАННЫХ ЛИДАРА ДЛЯ ВЫДЕЛЕНИЯ ОБЪЕКТОВ
ПРИ РЕШЕНИИ ЗАДАЧИ СЛЕДОВАНИЯ

© 2022 г. Р. Б. Рыбка¹, В. А. Шеин¹, М. С. Скороходов¹,
А. В. Грязнов¹, А. Г. Селиванов¹, А. Г. Сбоев^{1,2,*}

¹ Национальный исследовательский центр «Курчатовский институт», Москва, Россия

² Национальный исследовательский ядерный университет «МИФИ», Москва, Россия

*E-mail: Rybkarb@gmail.com

Поступила в редакцию 15.03.2022 г.

После доработки 20.03.2022 г.

Принята к публикации 20.03.2022 г.

Важной частью в задачах управления движением является определение препятствий вокруг агента следования. Для этого необходимо выделить точки поверхности из всего облака точек и удалить их. Представлено сравнение методов сегментации поверхности применительно к задаче выделения препятствий с бортового лидара, установленного на роботе. Проведены эксперименты по оценке точности выделения точек поверхности в типовых ситуациях функционирования робота, таких как попадание в углубление в рельефе, нахождение среди большого количества препятствий и нахождение на пустом и ровном участке поверхности. Приведена статистика при прохождении роботом одного маршрута с наличием всех типовых ситуаций. Исследованы распространенные методы выделения точек поверхности местности с данных лидара, такие как Random Sample Consensus (RANSAC), Progressive Morphological Filter (PMF), The Simple Morphological Filter (SMRF), Skewness Balancing и The Cloth Simulation Filter (CSF), реализованные в открытом программном обеспечении PDAL. Продемонстрированы особенности работы методов в типовых ситуациях функционирования агента на основе данных бортового лидара. В проведенном исследовании лучшие результаты показали методы SMRF и CSF.

DOI: 10.56304/S2782375X22030068

ВВЕДЕНИЕ

С помощью беспилотных средств можно исключить использование человеческих ресурсов в областях, которые несут высокий риск жизни и здоровью, например в пожаротушении, исследованиях в сложных и опасных условиях и т.д.

Использование робототехнических комплексов в труднодоступных регионах актуализирует задачу разработки алгоритмов управления, в частности, для случая движения агента за целевым объектом (ведущим) или движения в колонне в условиях динамически изменяемого маршрута с учетом возникающих препятствий.

При разработке систем управления важной частью является сенсорная система. В современных транспортных средствах широко используются датчики, такие как лидары и камеры.

Обнаружение и определение дальности при помощи света (LiDAR) — это метод дистанционного зондирования, использующий свет для измерения расстояния. Лидар в последнее время стал преобладать в транспортной инженерии, его

можно использовать, например, для сбора данных о дорожном движении, наличии препятствий и т.д. Бортовой лидар, также называемый мобильным лидаром, применяется для автоматизированного извлечения дорожной разметки [1–3], обнаружения тротуара [4], идентификации пешехода [5], обнаружения границ дороги [6] и отслеживания транспортных средств [7] на автономном транспортном средстве [8].

Один кадр данных лидара, называемый облаком точек, содержит десятки тысяч точек, включающих в себя информацию как о рельефе, так и об объектах. Лидар обнаруживает окружающие объекты, которые затем формируются в облако точек с координатами x, y, z на высокой частоте (обычно 5–20 Гц, варьируется у разных производителей LiDAR). Данные облака точек, полученные с помощью лидара, также использовались в цифровом моделировании автомагистралей [9], оценке аварийности городских дорог [10], обнаружении и классификации целей [11]. Информация о рельефе поверхности имеет решающее зна-

чение в обработке лазерного облака точек. Однако большое количество точек рельефа снижает скорость передачи и обработки данных и влияет на классификацию и распознавание основных целей [12, 13]. Таким образом, идентификация точек рельефа поверхности является ключевой при обработке всего облака точек [14]. Информация о рельефе поверхности, извлекаемая с помощью этой технологии, играет жизненно важную роль в планировании пути и улучшении точности распознавания целей. Мобильный лидар работает как источник локальных координат, он передает и принимает лазерный сигнал, в результате которого формируется массив данных с расположением множества точек вокруг с координатами x, y, z . Данные точки характеризуют расположение какого-либо объекта относительно лидара [15].

В настоящей работе сравниваются актуальные классические методы выделения точек рельефа из облака точек лидара на основе экспериментов в типовых условиях функционирования робота. Приведено описание методов фильтрации (сегментации) рельефа, описаны реализация данных методов с использованием языка Python и открытого программного обеспечения и эксперименты для получения данных. Представлены методы оценки точности, результаты тестирования и их сравнение применительно к задаче выделения точек препятствий от точек рельефа поверхности из всего облака точек лидара. Приведены сравнение и выбор наилучшего метода для его использования при решении задачи следования за лидером на основе нейросетевой модели.

МЕТОДЫ

Метод на основе случайной выборки (Random Sample Consensus, RANSAC) – итеративный метод, с помощью которого выбираются произвольные точки из выборки (три точки для плоскости) и формируется на их основе плоскость. Далее с учетом пороговых значений, отражающих максимальное значение перепада высоты в выбранной области, проверяется каждая точка выборки на то, попадает ли она в плоскость с пороговыми значениями. В случае, если данная точка попадает в область, она записывается в список точек, характеризующих рельеф поверхности, если не попадает, она удаляется (относится к списку препятствий). Аналогичная операция, только для других произвольных точек, повторяется в цикле до тех пор, пока не закончатся итерации. В цикле сохраняется наилучший вариант с наибольшим количеством точек, попадающих в допустимую область. Если во время работы алгоритма появляется вариант лучше, то предыдущие значения поверхности заменяются этим лучшим вариантом.

Алгоритм выглядит следующим образом.

Шаг 1. Выбор случайного набора точек (три точки для плоскости).

Шаг 2. Расчет параметров, необходимых для уравнения плоскости.

Шаг 3. Расчет отклонения всех точек в облаке точек от плоскости, используя оценку расстояния.

Шаг 4. Если расстояние находится в пределах заданного порогового значения отклонения от выбранной плоскости, добавить точку в список точек данной плоскости.

Шаг 5. Сохранение точек плоскости и списка точек с их максимальным количеством.

Шаг 6. Повторение процесса еще раз, пока не будет достигнуто максимальное количество итераций.

После завершения обработки результатом будет плоскость с максимальным количеством точек, которая является наилучшей оценкой базовой плоскости, т.е. плоскость земной поверхности. Любое скопление точек над этой плоскостью земли может быть классифицировано как объект, препятствие или дорожный знак.

Преимущества алгоритма заключаются в простоте его реализации и скорости работы, а также в способности дать качественный результат для выявления поверхности рельефа даже в случаях наличия большого количества препятствий.

Прогрессивный морфологический фильтр (Progressive Morphological Filter, PMF) – фильтр для сегментации рельефа поверхности, который является реализацией метода, описанного в [17]. Это итеративный метод, основанный на морфологических операциях фильтрации изображения. Математическая морфология в контексте машинного зрения – это описание свойств формы областей на изображении. В основе метода лежит морфологическая операция открытия, построенная на основе дилатации и эрозии.

Дилатация (морфологическое расширение) – свертка изображения или выделенной области изображения с некоторым ядром. Ядро может иметь произвольную форму и размер. При этом в ядре выделяется единственная ведущая позиция, которая совмещается с текущим пикселем при вычислении свертки. Во многих случаях в качестве ядра выбирается квадрат или круг с ведущей позицией в центре. Ядро можно рассматривать как шаблон или маску. Применение дилатации сводится к проходу шаблоном по всему изображению и применению оператора поиска локального максимума к интенсивностям пикселей изображения, которые накрываются шаблоном. Такая операция вызывает рост светлых областей на изображении (рис. 1в). На рисунке серым цветом отмечены пиксели, которые в результате применения дилатации будут белыми.

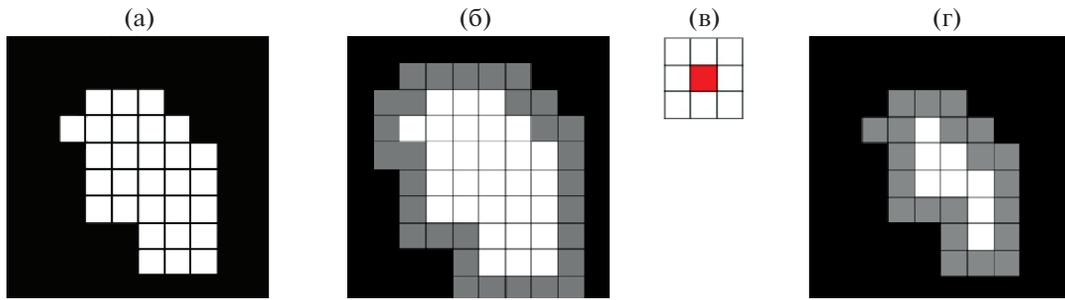


Рис. 1. Пример морфологических операций: а – исходное изображение, б – результат дилатации, в – шаблон (центр – ведущий элемент), г – результат эрозии.

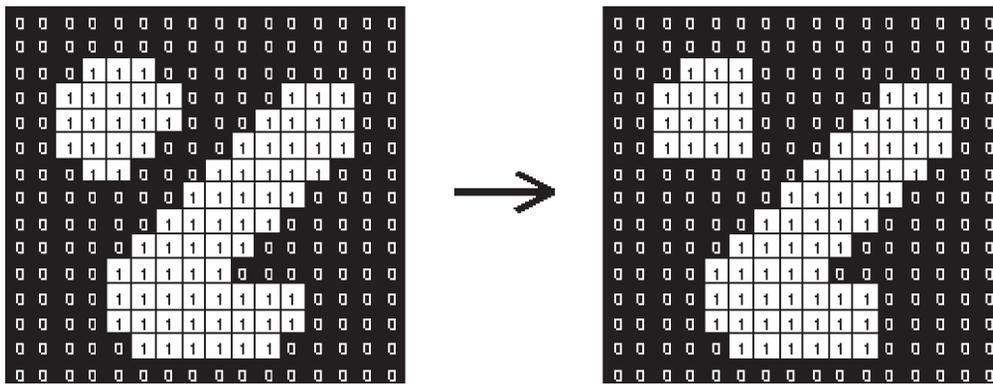


Рис. 2. Пример операции открытия на бинарном изображении.

Эрозия (морфологическое сужение) – обратная операция. Действие эрозии подобно дилатации, разница лишь в том, что используется оператор поиска локального минимума (рис. 1г), серым цветом залиты пиксели, которые станут черными в результате эрозии.

Операция открытия более мягко удаляет точки из исходного изображения в сравнении с эрозией и работает следующим образом: сначала к исходному изображению применяется операция эрозии с заданным ядром свертки, затем к полученному изображению применяется операция дилатации с тем же ядром (рис. 2).

Данные концепции распространились в области анализа непрерывной поверхности с использованием облака точек лидара.

Алгоритм заключается в следующем.

Шаг 1. Загружаются неравномерно расположенные (x, y, z) точки лидара. Регулярная минимальная сетка поверхности строится путем выбора точки с минимальным значением высоты (т.е. координаты с минимальным значением координаты z) в каждой ячейке сетки заданного размера. Координаты точки (x, y, z) хранятся в каждой ячейке сетки. Если ячейка не содержит измерений, ей присваивается значение измерения ближайшей точки.

Шаг 2. Прогрессивный морфологический фильтр, основной компонент – операция открытия – применяется к поверхности сетки. На первой итерации минимальная поверхность возвышения вместе с начальным размером окна фильтрации предоставляет входные данные для фильтра. В следующих итерациях отфильтрованная поверхность, полученная на предыдущей итерации, и увеличенный размер окна из шага 3 используются в качестве входных данных для фильтра. Результат этого шага включает в себя дальнейшее сглаживание рельефа от морфологического фильтра и обнаруженные точки объектов на основе порогового значения перепада высот.

Шаг 3. Размер окна фильтра увеличивается и вычисляется пороговое значение перепада высот. Шаги 2, 3 повторяются до тех пор, пока размер окна фильтра больше предварительно заданного максимального значения. Это значение обычно устанавливается равным немного больше, чем максимальный размер здания.

Шаг 4. Последний шаг – создание цифровой модели рельефа на основе полученного набора данных после того, как неземные точки объектов были удалены.

На вход алгоритма поступают значения координат точек лидара. Данные точки формируются

в сетку. Далее по сетке точек лидара осуществляется проход с минимальным окном (ядром). Перед запуском алгоритма задается максимальное значение окна, которое желательно выбирать с максимальными размерами зданий и крупных объектов. Далее запускается алгоритм.

На вход поступают точки (x, y, z) лидара и формируется сетка из минимальных значений для каждой ячейки сектора. Берется минимальное значение окна и применяется морфологический алгоритм. Окно (ядро) проходит по области точек, применяя операцию открытия. Результатом являются выделенная модель поверхности и неземные точки. В последующих итерациях отфильтрованная поверхность используется в качестве входных данных для фильтра. Выделение неземных точек осуществляется за счет проверки значения порога перепада высоты.

Данный алгоритм повторяется, пока размер окна не превысит размер максимального окна, который был задан заранее.

В результате работы алгоритма можно получить значения точек, относящихся к поверхности.

Простой морфологический фильтр (The Simple Morphological Filter, SMRF) классифицирует точки поверхности на основе подхода, изложенного в [18].

Алгоритм состоит из четырех этапов. На первом формируется минимальная поверхность, которая строится путем выбора точки с минимальным значением высоты в каждой ячейке сетки заданного размера по аналогии с алгоритмом РМФ. Координаты точки $(x; y; z)$ хранятся в каждой ячейке сетки. Если ячейка не содержит измерений, ей присваивается значение измерения ближайшей точки. Ячейки обозначают либо голую поверхность, либо объекты. На втором этапе применяется РМФ-алгоритм, описанный выше, в результате чего формируется бинарная сетка из ячеек поверхности и объектов. На третьем создается цифровая модель поверхности на основе бинарной сетки. Последним этапом алгоритма является идентификация исходных точек LiDAR земли/объектов в соответствии с цифровой моделью поверхности. Это достигается путем вычисления расстояния по высоте между каждой точкой LiDAR и предварительной цифровой моделью рельефа с учетом значения порога перепада высоты.

Алгоритм выглядит следующим образом.

Шаг 1. Создание сетки поверхности с сохранением всех точек лидара в соответствии с ячейками сетки заданного размера.

Шаг 2. Создание копии сетки поверхности (*last surface*). Наименьшая высота всех точек LiDAR в каждой ячейке сохраняется.

Шаг 3. Создание вектора размера окна на основе предоставленного максимума. Вектор увеличивается с одного пикселя на один пиксель до предела заданного максимального значения, деленного на размер ячейки.

Шаг 4. Для каждого размера окна:

4.1. вычисляется пороговое значение высоты, равное заданному параметру уклона, умноженному на произведение этого размера окна и размера ячейки минимальной сетки поверхности;

4.2. создается новая сетка поверхности (*this_surface*) в результате морфологического открытия, примененного к предыдущей сетке поверхности (*last_surface*), с помощью элемента формы диска с радиусом, равным текущему размеру окна;

4.3. добавляется к набору отмеченных ячеек земли любая ячейка, для которой разница между предыдущей сеткой поверхности (*last_surface*) и новой сеткой поверхности (*this_surface*) больше, чем пороговое значение высоты, рассчитанное в 4.1;

4.4. замена прошлой сетки поверхности на новую.

Шаг 5. Нахождение низких выбросов в сетке поверхности инвертированием поверхности и применением шагов 2, 3 для одного маленького окна и большого порога высоты.

Шаг 6. Создание предварительной цифровой модели рельефа. Из ячеек сетки сохраняются только ячейки, идентифицированные как рельеф, а пустые закрашиваются.

Шаг 7. Классификация точек поверхности. Исходные точки LiDAR классифицируются как рельеф на основе требуемого параметра вертикального расстояния и дополнительного параметра масштабирования.

Метод балансировки асимметрии (Skewness Balancing classifies) — это метод классификации точек рельефа, основанный на подходе [19].

Поскольку эксцесс и асимметрия выражают характеристики распределения облака точек, их можно в равной степени рассматривать как критерии завершения в алгоритме сегментации, т.е. если при вычислении асимметрии и эксцесса значение больше или равно нулю, алгоритм завершает удалять точки из всего облака, а оставшиеся точки будут характеризовать точки поверхности. В этом алгоритме неконтролируемой сегментации в качестве меры выбрана асимметрия. Поэтому алгоритм сегментации называется балансировка асимметрии и работает следующим образом.

Сначала вычисляется асимметрия облака точек:

$$sk = \frac{1}{N \cdot \sigma^3} \cdot \sum_{i=1}^N (s_i - \mu_a)^3, \quad (1)$$

Таблица 1. Меры распределения

Характеристика распределения	Преобладание внеземных точек	Преобладание точек рельефа	Нормальное распределение
Коэффициент асимметрии	$sk > 0$	$sk < 0$	$sk = 0$
Коэффициент эксцесса	$ku > 3$	$ku < 3$	$ku = 3$

$$ku = \frac{1}{N \cdot \sigma^4} \cdot \sum_{i=1}^N (s_i - \mu_a)^4, \quad (2)$$

где N – общее количество точек s_i лидара, σ – стандартное отклонение, μ_a – среднее арифметическое.

Если значение больше нуля, в облаке точек преобладают внеземные точки (пики/точки на большом расстоянии от поверхности земли) распределения (табл. 1). В этом случае самое высокое значение облака точек удаляется путем его классификации как точка объекта. Чтобы разделить все точки земли и объекта, эти шаги выполняются итеративно, в то время как асимметрия облака точек больше нуля. Наконец, оставшиеся точки в облаке принадлежат земле.

Алгоритм выглядит следующим образом.

Шаг 1. Загрузка значений точек лидара.

Шаг 2. Расчет значений асимметрии. Если значение больше нуля, сохраняются значения наивысших точек как точек объектов и удаляются из исходного облака.

Шаг 3. Повторение второго шага, пока значение асимметрии не станет меньше или равно нулю.

Шаг 4. Сохранение значения точек рельефа, когда значение асимметрии станет меньше или равно нулю.

Метод фильтрации на основе имитации ткани. Метод имитации ткани (*Cloth Simulation Filter, CSF*) – это метод извлечения рельефа из данных

LiDAR, основанный на моделировании физического процесса, который называется моделирование ткани [20].

Первоначально исходное облако точек инвертируют, т.е. переворачивают вверх ногами, а затем на перевернутую поверхность сверху “падает” ткань. Анализируя взаимодействия между узлами ткани и соответствующими точками LiDAR, можно определить окончательную форму ткани и использовать ее в качестве основы для классификации исходных точек на земные и точки объектов (рис. 3).

Ткань моделируют как сетку, состоящую из частиц с массой и взаимосвязями. Частица имеет постоянную массу, а взаимосвязь моделируется как “виртуальная пружина” и подчиняется закону Гука:

$$m \frac{\partial X(t)}{\partial t^2} = F_{ext}(X,t) + F_{int}(X,t), \quad (3)$$

где X – положение частицы в каждые моменты времени t , F_{ext} – внешняя сила, состоящая из сил гравитации и столкновения, F_{int} – внутренние силы, характеризующие взаимосвязь частиц.

Для фильтрации точек LiDAR были внесены модификации. Силы, действующие на частицу, рассматриваются как два дискретных шага. Перемещение каждой частицы вычисляется только от силы тяжести. То есть уравнение (3) решается с внутренними силами, равными нулю. Тогда

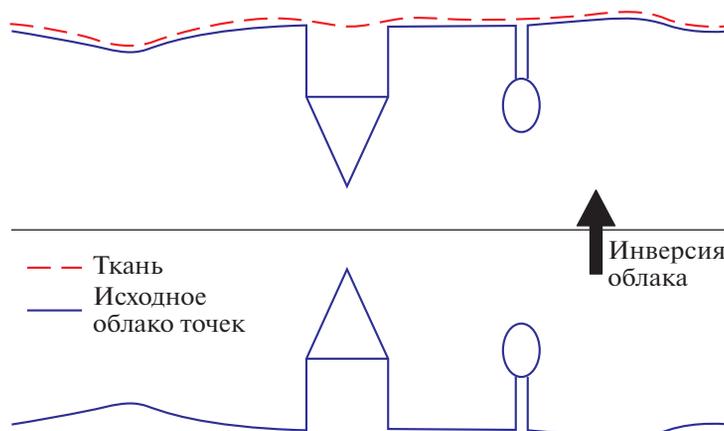


Рис. 3. Схема работы имитации ткани.

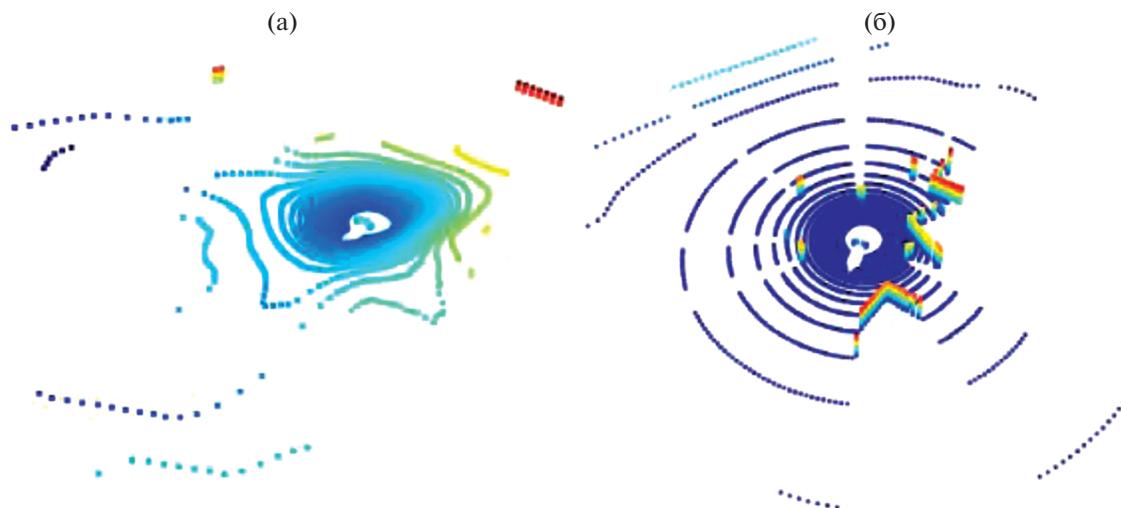


Рис. 4. Примеры облака точек ситуаций нахождения агента в углублении в рельефе (а) и среди большого количества препятствий (б).

формула интегрирования данного уравнения имеет вид

$$X(t + \Delta t) = 2X(t) - X(t - \Delta t) + \frac{G}{m} \Delta t^2. \quad (4)$$

Кратко алгоритм можно записать следующим образом.

Сначала проецируются частицы ткани и точки облака лидара на одну и ту же горизонтальную плоскость, затем находится ближайшая точка (названная соответствующей точкой (СТ)) для каждой частицы ткани в этой 2D-плоскости. Значение высоты пересечения (ЗВП) определяется для записи значения высоты (до проецирования) с СТ. Это значение представляет самое низкое положение, которого может достичь частица (т.е. если частица достигает самого нижнего положения, определяемого этим значением, она больше не может двигаться вперед). На каждой итерации сравниваем текущее значение высоты (ТЗВ) частицы с ЗВП; если ТЗВ равно или ниже ЗВП, возвращаем частицу в положение ЗВП и делаем частицу неподвижной.

После моделирования получается аппроксимация реальной местности, а затем расстояния между исходными точками облака лидара и смоделированными частицами рассчитываются с использованием алгоритма вычисления расстояния от облака к облаку [21]. Точки LiDAR с расстояниями меньше порогового значения (ПЗ) классифицируются как рельеф (голая земля), а остальные точки – как объекты.

Важной частью работы при решении задачи следования на робототехнической платформе является реализация данных методов на языке Python с использованием готовых решений из открытого доступа.

В результате анализа открытых источников было выделено несколько наиболее распространенных библиотек для работы с облаком точек, такие как Point Cloud Library (PCL) [22], Pdal [23], Open3D [24]. Все эти библиотеки имеют открытый исходный код.

Библиотеки Open3D и PCL имеют большое количество инструментов по работе с данными, однако в них реализован только простейший метод фильтрации RANSAC. Остальные методы, использованные в данном исследовании, были построены на базе библиотеки Pdal, имеющей большое количество готовых методов не только по обработке данных, но и фильтрации, сегментации рельефа, кластеризации объектов и другие, в результате чего данная библиотека стала основной. Библиотека Open3D использовалась в качестве предобработки исходного облака точек, а также для метода RANSAC.

Система управления движением на базе RL-модели. Для определения наилучшего метода выделения точек рельефа поверхности провели эксперименты в ситуациях, которые считаются типовыми для функционирования агента следования в условиях Арктики. В качестве исходных данных использовали 3D-среду функционирования агента следования, реализованную в Gazebo. В данной среде можно выделить распространенные ситуации, такие как пустые поверхности с уклонами, поверхности с углублениями и места с большим количеством препятствий. На основе этого выбрали следующие типовые ситуации для тестирования методов сегментации рельефа поверхности. Ситуация 1 (рис. 4а) – агент находится в углублении в рельефе. Ситуация 2 (рис. 4б) – агент находится на ровной поверхности среди большого количества препятствий вокруг. Ситуа-

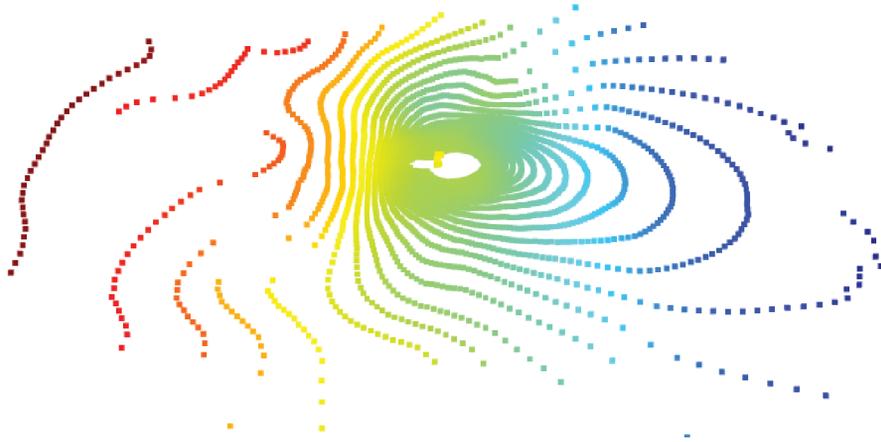


Рис. 5. Пример облака точек ситуации нахождения агента на ровной поверхности.

ция 3 (рис. 5) – агент находится на ровной поверхности с уклоном без наличия препятствий вокруг.

На рисунках до фильтрации синим отмечены низшие точки, а красным – высшие, после фильтрации черным цветом показаны точки поверхности, а фиолетовым – точки, относящиеся к объектам. Дальность точек от центра находится в радиусе 25 м. Для расчета использовали процессор Intel Core i7 третьего поколения.

Основной критерий качества заключается в том, чтобы после сегментации в оставшемся массиве точек не оставалось точек, которые будут находиться вблизи агента и относиться к рельефу поверхности. В каждой ситуации приводится таблица с результатами, а также наглядные примеры особенностей работы методов.

Для оценки точности работы методов выбрали типовые ситуации. Они были фиксированы, и эксперименты проводили приблизительно в одних заданных точках. В каждой ситуации проводили несколько запусков по получению исходного облака точек с лидара. Полученное облако с каждого запуска в каждой ситуации сегментировали вручную для получения значений количества точек рельефа и объектов. На основе данных значений проводили оценку точности работы методов.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Система оценки работы методов сегментации рельефа используется исходя из поставленной задачи. Для осуществления движения агента с использованием RL-модели необходимо с помощью абстрактных лучей проверять наличие препятствий на заданном расстоянии от агента. Длина лучей может варьироваться от 4 до 10 м. Обнаружение препятствий осуществляется с помощью обработки данных с лидара. Необходимо

отфильтровать максимальное количество точек рельефа, чтобы оставить все точки, характеризующие препятствия. На основе оставшихся точек будет осуществляться проверка наличия препятствия вблизи агента. Оценку работоспособности методов проводили после настройки параметров, значения которых обеспечивают наилучший результат в данной среде.

В качестве параметров для анализа эффективности работы методов рассматривали время работы каждого метода. Также оценивали точность работы методов фильтрации, которая основывается на формулах, предложенных в [25]. Выделены два типа ошибок. Ошибка типа 1 (ошибка поверхности) представляет процент точек поверхности, которые неправильно классифицируются как точки объектов, рассчитывается по формуле

$$\text{Type error 1} = \frac{b}{a + b}, \quad (5)$$

где a – количество правильно классифицированных точек поверхности, b – количество точек поверхности, ошибочно классифицированных как точки объектов.

Ошибка типа 2 (ошибка препятствий) представляет процент точек объектов, неправильно классифицированных как точки поверхности, рассчитывается по формуле

$$\text{Type error 2} = \frac{c}{c + d}, \quad (6)$$

где c – количество точек объектов, ошибочно классифицированных как точки поверхности, d – количество правильно классифицированных точек объектов.

Все эксперименты рассматривали визуально для понимания особенностей результатов. В качестве тестовых данных выбрали три типовые ситуации, которые являются самыми распространенными в среде функционирования агента.

Таблица 2. Результаты работы методов в углублении в рельефе

	Предобработка	RANSAC	PMF	Skewness balancing	SMRF	CSF
Время, с	0.046	0.057	0.013	0.010	0.017	0.013
Ошибка типа 1, %		0.27	4.82	19.90	0.02	0.11
Ошибка типа 2, %		4.00	42.26	0.00	11.60	1.66

Ключевым параметром из всех в поставленной задаче является оценка ошибки типа 1, так как при реализации задачи следования важно именно убрать все точки поверхности рельефа, даже если небольшое количество нижних точек препятствий тоже будет удалено.

Ситуация попадания в углубление в рельефе. В табл. 2 представлены результаты работы методов выделения точек поверхности. Так как в данной ситуации среди всех точек присутствует небольшое количество точек, относящихся к объектам, в таблице приведены два типа ошибок. В колонке “Предобработка” представлено среднее время предобработки перед осуществлением фильтрации. В остальных колонках представлены результаты работы методов.

При анализе результатов можно выделить следующие недостатки методов. Метод балансировки асимметрии плохо справляется с фильтрацией рельефа в случаях нахождения агента в локальных небольших ямах.

На рис. 6, 7 приведен пример сравнения со всеми точками. Это обусловлено тем, что при попадании в углубление рельефа самыми низшими точками являются точки этого углубления, которые и будут являться точками поверхности при вычислении асимметрии, а наивысшие точки классифицируются как объекты.

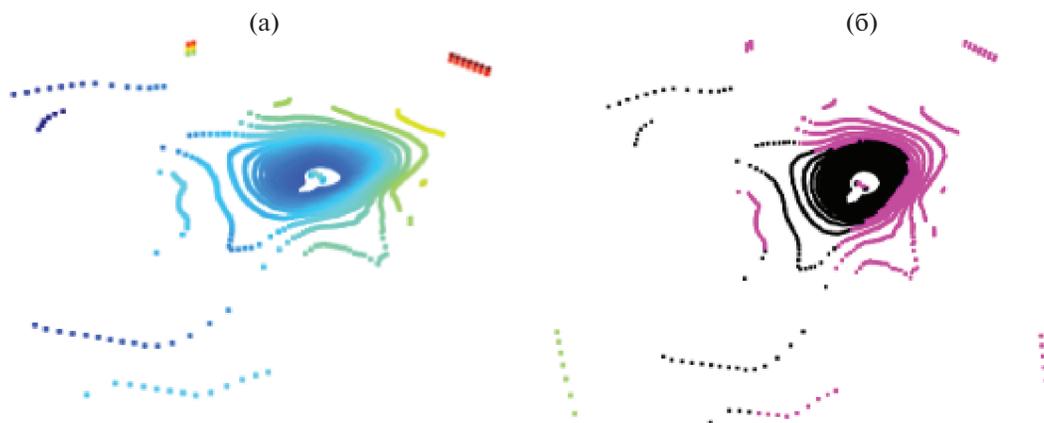
Аналогичным образом справляется с данной ситуацией прогрессивный морфологический

фильтр. Отметим, что по сравнению с методом балансировки асимметрии он справляется значительно лучше, однако при решении задачи следования применить данные методы нельзя, поскольку при разработке системы управления для объезда препятствий необходимо среди всех точек лидара выделить только те, которые относятся к препятствиям. В дальнейшем на базе данных точек формируется коридор, в котором можно осуществлять движения, а если точки рельефа попадают в список с препятствиями, то модель не сможет осуществить движение.

Наилучшие результаты в заданной ситуации продемонстрировали методы SMRF и CSF, у которых средняя ошибка выделения поверхности составляет меньше 0.1%, что говорит о высокой точности работы. Высокий процент ошибки типа 2 не говорит о том, что метод SMRF справляется плохо при выделении препятствий. Поскольку из-за небольшого количества точек, характеризующих объекты, при попадании даже небольшого количества точек нижние части объектов, которые находятся у самой земли, влияют на результаты ошибки типа 2.

Ситуация нахождения агента среди большого количества препятствий вокруг. В табл. 3 приведены сравнительные характеристики средних значений времени работы методов и количество точек после фильтрации.

При визуальном анализе выделяются основные проблемы при работе методов. Поверхность в

**Рис. 6.** Пример фильтрации методом балансировки асимметрии.

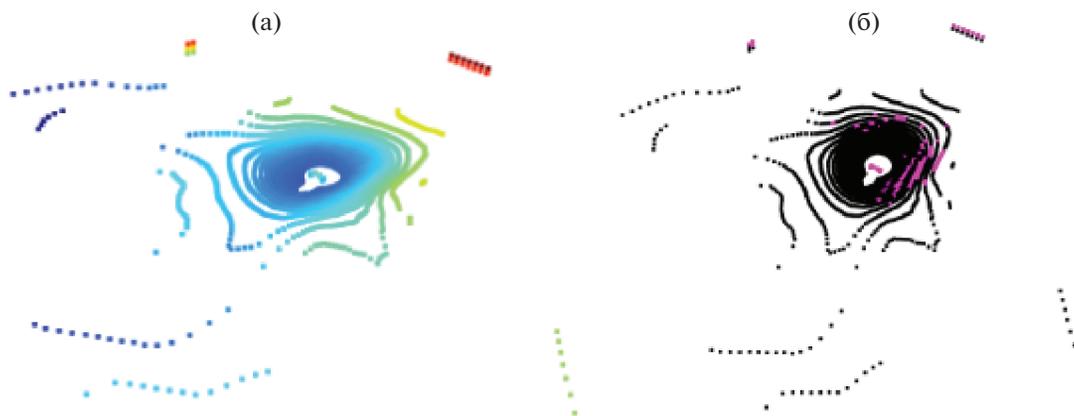


Рис. 7. Пример фильтрации прогрессивным морфологическим методом в условиях углубления.

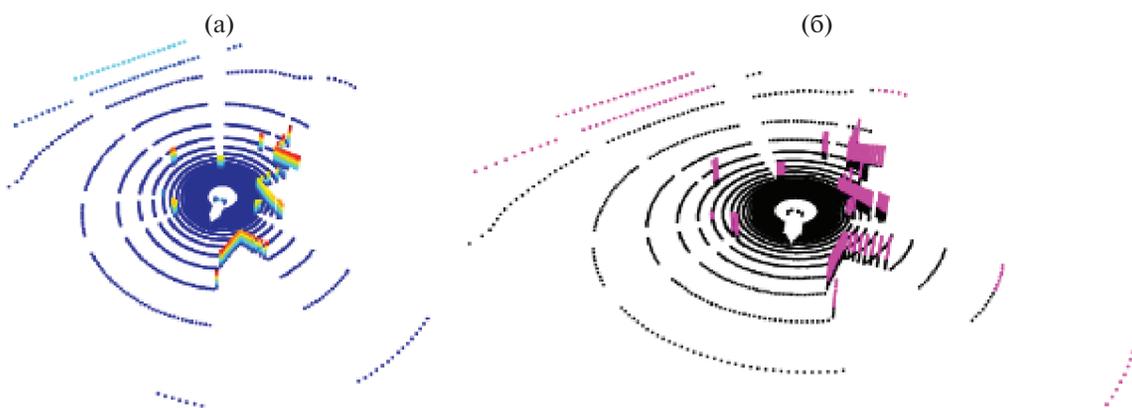


Рис. 8. Сравнение метода RANSAC со всеми полученными точками.

большей степени имеет ровный рельеф, однако, как видно на рис. 8, в верхней левой части имеется возвышенность, что видно в смене цвета точек с синего на голубой.

В силу того что RANSAC подбирает плоскость, то при большом количестве препятствий он не всегда может идеально подобрать ее даже при большом пороговом значении перепада высоты.

На рис. 8 представлен пример результата фильтрации методом RANSAC. В данном случае метод не отфильтровал только дальние точки, что можно было бы решить уменьшением радиуса обработки, однако данный метод не всегда обрабатывает и ближайшие точки поверхности. Также

можно заметить, что при большом пороговом значении захватывается большое количество точек препятствий, в силу чего маленькие препятствия могут быть отнесены к точкам поверхности, что плохо сказывается при управлении движением агента.

На рис. 9 представлено сравнение методов балансировки асимметрии и имитации ткани. Как можно заметить, они хорошо осуществляют фильтрацию, за исключением той области, где начинается возвышенность рельефа.

Полученные результаты не критичны, поскольку это решается уменьшением радиуса дальности обработки точек, что удовлетворяет по-

Таблица 3. Результат работы методов в ситуации с большим количеством препятствий

Средняя	Предобработка	RANSAC	PMF	Skewness balancing	SMRF	CSF
Время, с	0.053	0.098	0.054	0.010	0.018	0.075
Ошибка типа 1, %		1.86	0.00	2.00	0.00	0.54
Ошибка типа 2, %		39.20	8.75	5.00	14.10	26.82

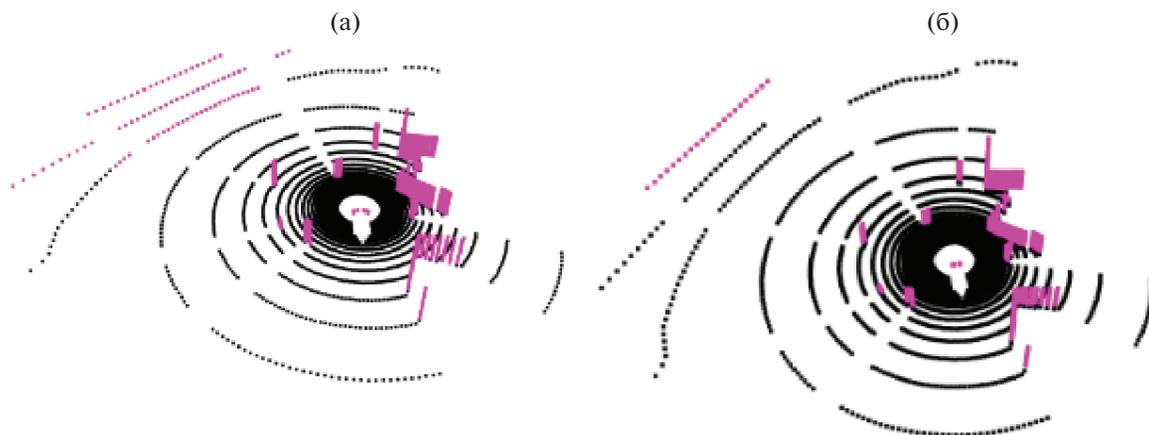


Рис. 9. Сравнение Skewness balancing и CSF.

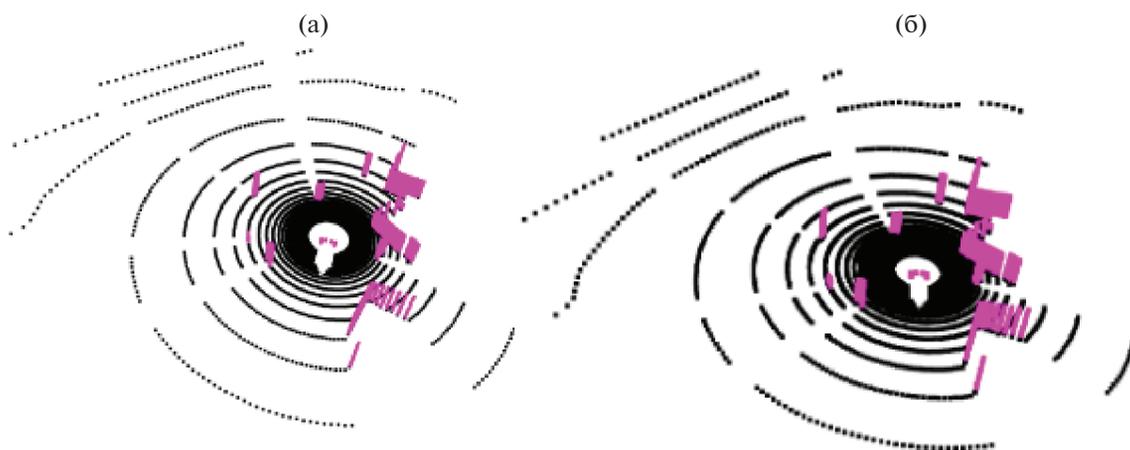


Рис. 10. Результаты работы методов PMF (а) и SMRF (б).

ставленной задаче. А при приближении к возвышенности эти методы уже справляются с данной проблемой при фильтрации.

Самые лучшие результаты показали простой и прогрессивный морфологические фильтры (рис. 10).

Так как обработка точек лидара и формирование списка с точками препятствий происходят каждый момент времени движения агента, использовать метод RANSAC в системе не рекомендуется из-за его низкой надежности. При большом пороговом значении метод захватывает нижние точки препятствий, а в данном случае низкие препятствия также могут быть отфильтрованы вместе с рельефом. Таким образом, метод имеет низкое качество обработки для данных задач.

Skewness balancing и CSF тоже показывают удовлетворительные результаты их применения в задаче выделения препятствий, однако в случаях с небольшим перепадом высот и наличием боль-

шого количества препятствий наилучшими методами являются PMF и SMRF.

Ситуация расположения агента на ровной плоскости с небольшим перепадом высоты. В табл. 4 представлены средние результаты работы методов и средний процент ошибок. Так как в данной ситуации отсутствуют препятствия, то ошибки типа 2 здесь нет.

Рассмотрим основные выявленные проблемы при проведении экспериментов. Метод балансировки асимметрии (рис. 11) в большинстве случаев полностью не фильтровал точки рельефа. Лишь в паре случаев он смог предоставить идеальные результаты фильтрации. Именно поэтому в табл. 4 число почти совпадает с общим числом точек, получаемых с лидара, средняя ошибка превышает 99%.

Прогрессивный морфологический фильтр в среднем выдавал неплохие результаты фильтрации, но после обработки порой оставались неотфильтрованные участки, где высота на окраине

Таблица 4. Результаты работы методов в ситуации на ровной плоскости

	Предобработка	RANSAC	PMF	Skewness balancing	SMRF	CSF
Время, с	0.026	0.053	0.009	0.018	0.015	0.012
Ошибка типа 1, %		0.25	2.85	99.52	0.00	1.14

Таблица 5. Результаты работы методов при прохождении маршрута

	RANSAC	PMF	Skewness balancing	SMRF	CSF
Среднее количество всех точек	2749	2796	2737	2731	2751
Среднее количество точек после фильтрации	56	173	933	82	106
Среднее время обработки данных и фильтрации	0.046	0.016	0.131	0.013	0.016
Время прохождения маршрута, с	103.2	87.8	не доехал	90.6	79.8

отличалась от изначальной точки (рис. 11). Иногда метод выдавал плохие результаты фильтрации вблизи агента из-за некорректной фильтрации и конечного соотношения точек. Пример представлен на рис. 12, где можно увидеть результат работы PMF. На рисунке видно, как ошибочно выделяются точки поверхности как точки объектов вблизи агента, что негативно сказывается на работе системы управления движением агента.

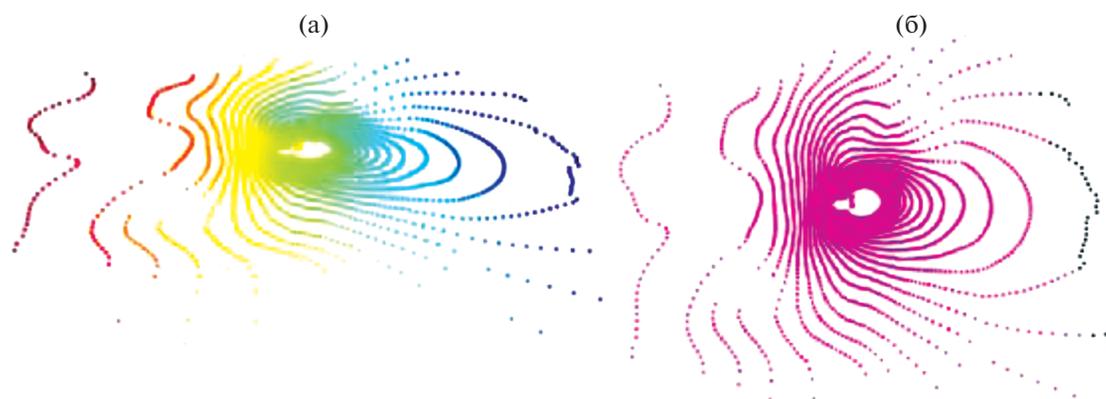
Именно из-за таких случаев среднее значение отфильтрованных точек такое, как указано в табл. 5.

Метод имитации ткани, показывавший хорошие результаты в предыдущих экспериментах, имеет аналогичный недостаток, захватываются дальние точки при наличии даже небольшого перепада. Однако он осуществляет отличную фильтрацию вблизи агента на расстоянии до 10–15 м (рис. 9). В среднем, проблема фильтрации была только на отдельных точках, что не влияет на работу системы управления. Но в большинстве случаев (~75%) метод CSF на ровной поверхности

дает идеальные результаты фильтрации, а также обладает высокой степенью быстродействия.

Лучшими в данном эксперименте оказались методы SMRF и RANSAC. Как и в прошлых экспериментах, RANSAC на ровной поверхности с большим пороговым значением может хорошо отфильтровать все точки. Алгоритм SMRF показывает прекрасный результат фильтрации облака точек лидара. Во всех случаях он четко фильтровал рельеф поверхности даже на неровных участках. Данный метод отличается быстродействием. Значения неотфильтрованных точек в табл. 5 характеризуют точки, относящиеся к самому агенту, и говорят о том, что методы RANSAC и CSF осуществляют идеальную фильтрацию, оставляя только те точки, которые не относятся к рельефу.

Из табл. 4 видно, что метод RANSAC имеет ошибку меньше 1%. На рис. 13 видно, как иногда попадают ошибочные точки, однако это все равно отличный результат работы. Алгоритм SMRF показал идеальные результаты, исходя из поставленной задачи.

**Рис. 11.** Сравнение всех точек с результатом работы метода балансировки асимметрии.

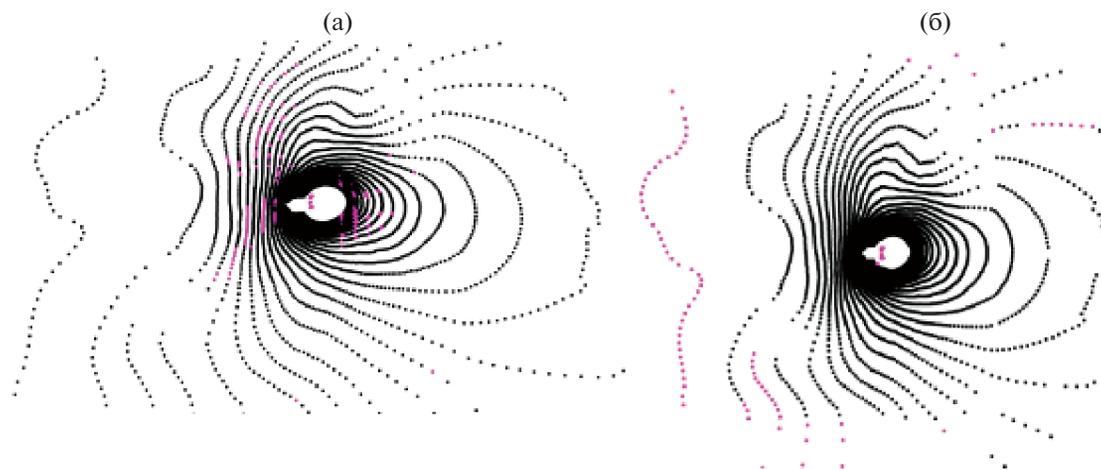


Рис. 12. Пример работы PMF (а) и CSF (б).

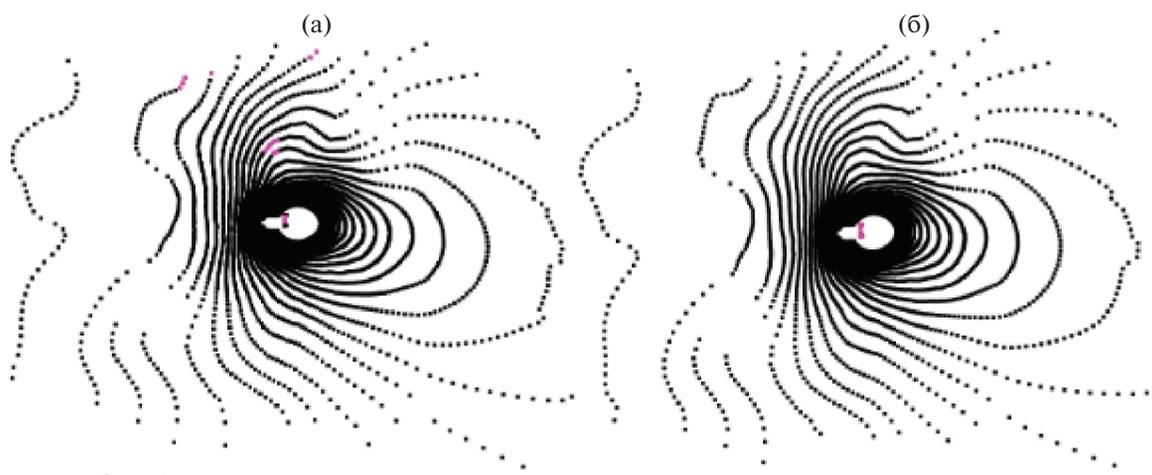


Рис. 13. Пример работы RANSAC (а) и SMRF (б).

Общая оценка каждого метода при движении агента по маршруту следования. Для общего тестирования выбрали один тестовый маршрут с небольшим количеством препятствий на пути и провели сравнение работы каждого метода посредством оценки основных показателей качества. Ключевым параметром здесь является время работы алгоритмов фильтрации и то, как оно влияет на качество прохождения маршрута. Основные результаты представлены в табл. 5.

Отметим, что помимо перечисленных параметров на способность агента успешно проходить маршрут влияет множество других параметров. Однако в табл. 5 отражены именно статистические данные при прохождении одного маршрута с разными алгоритмами фильтрации.

При проверке работоспособности метод Skewness Balancing показывает наихудшие результаты. С данным алгоритмом агент вообще не мог нор-

мально функционировать. Из табл. 5 по среднему значению точек после фильтрации видно, что метод балансировки асимметрии при прохождении маршрута очень плохо фильтрует рельеф относительно остальных методов.

Алгоритм PMF показал удовлетворительные результаты при прохождении маршрута, однако его слабая надежность часто сбивала агента с маршрута. Это видно из среднего количества точек после фильтрации, что значительно выше, чем у остальных методов.

Остальные методы показали отличные результаты, однако, используя RANSAC, агент в среднем проходил заданный маршрут дольше, чем при использовании других методов. Быстрее всего агент справился, применяя метод фильтрации CSF, однако наилучшие показатели в соотношении скорости и качества демонстрирует алгоритм SMRF. Данный метод обладает наилучшей скоро-

стью фильтрации, неплохой скоростью прохождения маршрута и высоким значением соотношения награды к общему числу шагов.

ЗАКЛЮЧЕНИЕ

Среди всех изученных методов наилучшие показатели продемонстрировал метод SMRF. Он не только наиболее точно фильтрует рельеф, но и показывает свое быстрое действие. Метод RANSAC имеет недостатки из-за грубой фильтрации по пороговым значениям. Задавая большое значение порога, можно добиться хорошего результата по выделению точек препятствий, но в редких случаях низкие препятствия под колесами агента могут быть отнесены к поверхности рельефа. Также данный метод имеет низкое быстрое действие, что не удовлетворяет требованиям при решении задачи.

Метод балансировки асимметрии хорошо справляется при большом количестве препятствий, но он совершенно не надежен. Он плохо справляется при попадании в углубление в рельефе, а также при движении по относительно ровной поверхности. Данный метод нельзя применить при решении задачи следования, поскольку неотфильтрованные точки рельефа в большом количестве являются помехой для RL-модели.

Метод PMF показал хорошие результаты, однако довольно часто на относительно ровных поверхностях и в углублениях в рельефе он может некорректно осуществлять фильтрацию. Такие сбои в целом не были критичными при прохождении маршрута, однако такие результаты периодически сбивают модель следования, что может выбить агента из коридора, в котором ему необходимо двигаться.

Хорошие результаты продемонстрировали методы имитации ткани и простой морфологический метод. Данные методы довольно хорошо справляются во всех случаях тестирования. При ограничении дальности видимости точек до 10–15 м они справляются практически в 100% случаев. Агент без проблем проходит маршрут, а мелкие ошибки фильтрации, которые случаются редко, не позволяют ему сбиться с маршрута. Данные методы надежны в различных условиях функционирования агента при движении в сложных участках рельефа, как в углублениях, так и в местах большого количества препятствий.

Именно простой морфологический метод (SMRF) и методы имитации ткани (CSF) взяты за основу для сегментации точек рельефа при решении задачи следования с помощью методов обучения с подкреплением.

Работа выполнена при поддержке НИЦ «Курчатовский институт» (приказ № 2754 от 28.10.2021).

СПИСОК ЛИТЕРАТУРЫ

1. *Ly B., Xu H., Wu J. et al.* // IEEE Access. 2019. V. 7. P. 76779.
2. *Yu Y., Li J., Guan H. et al.* // IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2015. V. 8. № 2. P. 709.
3. *Guan H., Li J., Yu Y. et al.* // IEEE Trans. Intell. Transp. Syst. 2015. V. 16. № 5. P. 2457.
4. *Ai C., Tsai Y.* // Transp. Res. Rec., J. Transp. Res. Board. 2016. V. 2542. № 1. P. 25.
5. *Ogawa T., Sakai H., Suzuki Y. et al.* // IEEE Intell. Vehicles Symp. 2011. V. 4. P. 734.
6. *Wijesoma W.S., Kodagoda K.S., Balasuriya A.P.* // IEEE Trans. Robot. Autom. 2004. V. 20. № 3. P. 456.
7. *Miyasaka T., Ohama Y., Ninomiya Y.* // IEEE Intell. Vehicles Symp. 2009. P. 151.
8. *Laugier C., Chartre J.* // Conference GTC Europe. 2016.
9. *Yang B., Fang L., Li J.* // ISPRS J. Photogrammetry and Remote Sensing. 2013. V. 79. P. 80.
10. *Guan H., Yu Y., Ji Z. et al.* // Remote Sens. Lett. 2015. V. 6 № 11. P. 864.
11. *Yan L., Li Z., Liu H. et al.* // Optics and Laser Technology. 2017. V. 97. P. 272.
12. *Huang S., Liu L., Dong J. et al.* // Eng. Comput. 2021. V. 38. № 4. P. 1895.
13. *Habermann D., Hata A., Wolf D. et al.* // IEEE. III Brazilian Symposium on Computing Systems Engineering. 2013. P. 143.
14. *Zhiqing L., Shenhua H., Qi M. et al.* // Geomatics Spatial InfoRMation Technology. 2019. V. 42 № 2. P. 40.
15. *Chu P.M., Cho S., Park J. et al.* // Human-centric Computing and Information Sciences. 2019. V. 9. № 1. P. 1.
16. *Nistér D.* // Mach. Vis. Appl. 2005. V. 16. № 5. P. 321.
17. *Zhang K., Chen S.C., Whitman D. et al.* // IEEE Trans. Geosci. Remote Sens. 2003. V. 41. № 4. P. 872.
18. *Pingel T.J., Clarke K.C., McBride W.A.* // ISPRS J. Photogrammetry Remote Sens. 2013. V. 77. P. 21.
19. *Bartels M., Wei H.* // Pattern Recognition Lett. 2010. V. 31. № 10. P. 1089.
20. *Zhang W., Qi J., Wan P. et al.* // Remote Sens. 2016. V. 8. № 6. P. 501.
21. *Girardeau-Montaut D.* // EDF R&D Telecom Paris-Tech. 2016. V. 11.
22. *Rusu R.B., Cousins S.* // IEEE Int. Conf. Robotics and Automation. 2011. P. 1.
23. PDAL Contributors. (2022). PDAL Point Data Abstraction Library.
24. *Zhou Q.Y., Park J., Koltun V.* // arXiv preprint arXiv:1801.09847. 2018.
25. *Moudry V., Klápště P., Fogl M. et al.* // Measurement. 2020. V. 150. P. 107047.