

64. Pshikhopov V.Kh., Medvedev M.Yu., Gaiduk A.R., Fedorenko R.V., Krukhmalev V.A., Gurenko B.V. Position-Trajectory Control System for Unmanned Robotic Airship, *IFAC Proceedings Volumes (IFAC-PapersOnline). The 19th World Congress the International Federation of Automatic Control. Cape Town, South Africa. August 24-29, 2014*, pp. 8953-8958.
65. Pshikhopov V.Kh., Medvedev M.Yu. Sintez sistem upravleniya podvodnymi apparatami s nelineynymi kharakteristikami ispolnitel'nykh organov [Synthesis of control systems for underwater vehicles with nonlinear characteristics of executive bodies], *Izvestiya YuFU. Tekhnicheskie nauki [Izvestiya SFedU. Engineering Sciences]*, 2011, No. 3 (116), pp. 147-156.

Статью рекомендовал к опубликованию к.т.н., с.н.с. В.А. Шевченко.

Медведев Михаил Юрьевич – НИИ робототехники и процессов управления Южного федерального университета; e-mail: medvmihal@sfedu.ru; г. Таганрог, Россия; тел.: 88634371694; д.т.н.; в.н.с.

Пшихопов Вячеслав Хасанович – e-mail: pshichop@rambler.ru; д.т.н.; профессор; директор.

Бросалин Дмитрий Олегович – e-mail: brosalin@sfedu.ru; инженер.

Гуренко Борис Викторович – e-mail: borisgurenko@sfedu.ru; к.т.н.; с.н.с.

Васильева Мария Александровна – Южный федеральный университет; e-mail: marv@sfedu.ru; г. Таганрог, Россия; тел.: 88634371694; магистрант.

Хамдан Низар – e-mail: dr.nizar.abou.hamdane@gmail.com; аспирант.

Medvedev Mikhail Yurjevich – R&D Institute of Robotics and Control Systems; e-mail: medvmihal@sfedu.ru; Taganrog, Russia; phone: +78634371694; dr. of eng. sc.; leading researcher.

Pshikhopov Viacheslav Khasanovich – e-mail: pshichop@rambler.ru; dr. of eng. sc.; professor; director.

Brosalin Dmitry Olegovich – e-mail: brosalin@sfedu.ru; engineer.

Gurenko Boris Viktorovich – e-mail: borisgurenko@sfedu.ru; cand. of eng. sc.; senior researcher.

Vasileva Maria Aleksandrovna – Southern Federal University; e-mail: marv@sfedu.ru; Taganrog, Russia; phone: +78634371694; student.

Hamdan Nizar – e-mail: dr.nizar.abou.hamdane@gmail.com; post-graduate student.

УДК 004.021

DOI 10.18522/2311-3103-2022-3-192-201

И.О. Шепель

ОБНАРУЖЕНИЕ ДИНАМИЧЕСКИХ ОБЪЕКТОВ НА КАРТЕ ЗАНЯТОСТИ С НАКОПЛЕНИЕМ НА ОСНОВЕ ФИЛЬТРА ЧАСТИЦ

Рассматривается проблема обнаружения динамических препятствий на карте занятости, полученной по данным системы технического зрения мобильной робототехнической платформы. Целью работы является качественное улучшение алгоритма обнаружения препятствий с помощью добавления фильтра частиц для выделения движущихся объектов по данным карты. В исследовании решается задача корректного накопления данных в карте занятости и уменьшения задержки обновления информации в ячейках карты, по которым двигаются динамические объекты. Представленная в статье модификация фильтра частиц способна корректно работать с динамическими препятствиями в широком диапазоне скоростей, устойчива к выбросам, вызванным в результате случайной генерации начальной скорости частиц, и предназначена для работы в реальных условиях в среде с большим количеством движущихся препятствий и в реальном масштабе времени. Разработана эвристика, которая уменьшает количество неправильных классификаций в окклюзированных зонах. Показано, что алгоритм обнаружения динамических объектов в

карте занятости инвариантен к типу сенсоров, используемых в системе технического зрения, а также описана реализация, объединенная с накапливаемой картой препятствий. Алгоритм реализован и протестирован на борту автономной робототехнической платформы и на открытом наборе данных. В статье приведено сравнение с другими подходами к обнаружению динамических препятствий, а также рассчитаны метрики быстродействия для всех анализируемых методов для вычислителей на базе GPU Nvidia RTX 3070 и GPU встроенного вычислителя Jetson AGX Xavier. Сформулированы перспективные направления дальнейших исследований по улучшению представленного подхода.

Карта занятости; лидар; облако точек; фильтр частиц; обнаружение препятствий; динамический объект; автономное движение.

I.O. Shepel

PARTICLE FILTER BASED DETECTION OF DYNAMIC OBJECTS ON AN ACCUMULATED OCCUPANCY MAP

The paper considers the problem of detecting dynamic obstacles on the accumulated occupancy map generated by the computer vision system of a mobile robot. The purpose of this research is to improve the quality of the obstacle detection algorithm by adding a particle filter to find moving objects from the map data. In the paper, the problem of correct accumulation of data in the occupancy map and reducing the delay in updating the map cells in which the object moves is solved. The modification of the particle filter presented in the paper is able to work correctly with dynamic obstacles in a wide range of speeds; it is resistant to outliers caused by random generation of the initial particles velocities, and is workable under real conditions in real time in an environment with a lot of moving objects. A heuristic has been created that reduces the number of misclassifications in occluded areas. It is shown that the algorithm for detecting dynamic objects in the map is invariant to the type of sensors used in the vision system, and an implementation combined with an accumulated occupancy map is described. The algorithm is implemented and tested on board an autonomous mobile robot, as well as on an open dataset. The article also provides a comparison with other approaches of dynamic obstacles detection, as well as calculated performance metrics for all analyzed methods for computers based on the GPU Nvidia RTX 3070 and Jetson AGX Xavier. Promising directions for further research to improve the presented algorithm are formulated.

Occupancy map; lidar; point cloud; particle filter; obstacle detection; dynamic object; autonomous movement.

Введение. В настоящее время активно развиваются и используются сервисные автономные робототехнические платформы (РТП) на базе небольших шасси, которые должны быть способны к безаварийному движению в пешеходных зонах с большим количеством движущихся объектов, например для решения задачи доставки последней мили или уборки улиц. Как правило, в подобных РТП используется классический подход к определению степени проходимости окружающей среды — построения карт занятости. Эти карты должны быть способны, во-первых, накапливать и уточнять информацию о препятствиях, а во-вторых, обновлять информацию о местоположении наиболее опасных динамических объектов с минимальными задержками. Наиболее распространенные подходы к обнаружению динамики либо ориентированы на применение на беспилотных автомобилях с другим доменом целевых данных, либо требуют больших вычислительных ресурсов, использование которых невозможно на небольших и экономичных сервисных РТП. Кроме того, пешеходные зоны характеризуются большой плотностью и разнообразностью движущихся объектов (дети, велосипеды, самокаты, животные), хаотичным характером движения и отсутствием разметки, что увеличивает сложность задачи обнаружения динамических объектов.

В работе рассматривается задача построения карты проходимости на борту автономной РТП в среде с большим количеством динамических объектов для обеспечения безаварийного движения при выполнении задач уборки территории и

курьерской доставки. Используемый подход позволяет нивелировать задержки в обновлении статуса ячеек с движущимися препятствиями в накапливаемой карте проходимости, что позволяет оперативно реагировать на изменение их позиции и в дальнейшем корректировать собственную траекторию движения РТП.

В статье описана модификация алгоритма построения накапливаемой карты занятости по облаку точек с использованием фильтра частиц. Ключевые отличия разработанного алгоритма следующие:

- ◆ реализованный алгоритм работоспособен в реальном масштабе времени на встраиваемых вычислительных блоках;
- ◆ разработана эвристика, повышающая устойчивость алгоритма к ложным детекциям динамических ячеек;
- ◆ разработана эвристика, уменьшающая влияние окклюдированных зон на ложные детекции динамических препятствий;
- ◆ разработанный алгоритм имплементирован в алгоритм построения накапливаемых карт, что позволяет обнаруживать динамические объекты по нескольким различным источникам сенсорных данных.

Обзор существующих методов. Карта занятости [1] является одним из самых распространенных методов для построения модели проходимости и решения задачи беспилотного движения РТП. При этом большинство подходов не решает задачу выделения и отдельной обработки ячеек движущихся препятствий. Для обнаружения динамики, как показывают открытые конкурсы [2, 3], наиболее успешными являются нейросетевые подходы [4, 5]. Такие методы требуют высокопроизводительных вычислителей и больших наборов данных нужного домена, что является проблемой при разработке системы обнаружения препятствий в сервисных РТП. Планировать траекторию по таким данным без дополнительной картографической информации нельзя, но в работе [6] показана принципиальная возможность комплексировать такие данные с картами проходимости. Другой подход к выделению потенциальных динамических препятствий также использует нейросетевые модели, но для решения задачи семантической сегментации, как показано в [7]. При этом, хотя такой подход реализуем как на данных от стереокамер [8, 9], так и на облаках точек от лидара [10, 11], его использование не на беспилотных автомобилях затруднительно из-за отсутствия открытых наборов данных для обучения. Кроме того, вышеперечисленные методы существенно увеличивают время обнаружения объектов в связи с использованием тяжелых моделей. Другое семейство алгоритмов для обнаружения динамики также использует сверточные сети, но в качестве входа подаются классические карты занятости [12, 13]. Для обнаружения и предсказания движения используются рекуррентные сети, но такие модели очень чувствительны к параметрам карты занятости, что затрудняет обучение модели, применимой в реальных условиях. Наиболее универсальными являются подходы, не использующие нейронные сети, в частности базирующиеся на алгоритме фильтрации частиц [14, 15]. Поскольку частицы в них генерируются случайным образом, то точность обнаружения напрямую зависит от количества этих частиц, а значит и от производительности вычислителя, но существуют параллельные реализации, решающие эту проблему [16].

Алгоритм обнаружения динамики на накапливаемой карте занятости. В работе рассматривается проблема выделения движущихся объектов на карте занятости, сгенерированной по данным от лидара. Представленный алгоритм является развитием работ автора по комплексированию данных системы технического зрения РТП и обнаружению динамики [6, 7] и построению карты проходимости от нескольких сенсоров [17]. В качестве базовой реализации алгоритма фильтрации частиц был выбран метод из [16]. В нем можно выделить следующие шаги:

1. Предсказание положения частиц.
2. Распределение частиц по ячейкам.
3. Расчет обновленной вероятности занятости ячеек.
4. Обновление весов частиц с предыдущего шага.
5. Инициализация новых частиц.
6. Вычисление среднего и дисперсии двухмерной скорости частиц.
7. Повторная выборка и отсеивание частиц.
8. Выделение динамических ячеек карты по ранее найденным метрикам.

Предсказание положения и распределение частиц. Частицы представляют собой вектор вида $p = (x, y, v_x, v_y)$, при этом их модель движения считается линейной. На шаге предсказания частицы перемещаются по карте в соответствии со своими скоростями, а затем проецируются в ячейки карты. Вес сохранившихся с предыдущего шага частиц обновляется по формуле (1):

$$\omega^{i,p} = p_S \cdot \omega^i, \quad (1)$$

где p_S – гиперпараметр алгоритма, описывающий вероятность «выживания» частицы.

Расчет обновленной вероятности занятости ячеек. На этом шаге рассчитываются массы гипотез «ячейка занята» $m^{i,p}(O^{i+1})$ и «ячейка свободна» $m^{i,p}(F^{i+1})$:

$$\begin{aligned} m^{i,p}(O^{i+1}) &= \sum_{k=1}^{K_P} \omega_k^{i,p}, \\ m^{i,p}(F^{i+1}) &= \min[m^i(F^i), 1 - m^{i,p}(O^{i+1})], \end{aligned} \quad (2)$$

где K_P – число частиц в ячейке. Формулы (2) используются для вычисления присоединенной массы по правилу комбинации Демпстера-Шафера [18] (3):

$$m^{i+1}(O^{i+1}) = m^{i,p}(O^{i+1}) \oplus m^i(O^{i+1}|z^{i+1}). \quad (3)$$

Масса $m^i(O^{i+1}|z^{i+1})$ получена с моментальной карты измерений, как это показано в [17]. Аналогичная формула используется и для свободных ячеек. Суммарная масса частиц в ячейке на каждой итерации состоит из «выживших» и «рожденных» частиц. Эти массы представимы в виде (4):

$$\begin{aligned} \rho^{i+1,b} &= \frac{m^{i+1}(O^{i+1}) \cdot p_B [1 - m^{i,p}(O^{i+1})]}{m^{i+1}(O^{i+1}) - p_B [1 - m^{i,p}(O^{i+1})]}, \\ \rho^{i+1,p} &= m^{i+1}(O^{i+1}) - \rho^{i+1,b}, \end{aligned} \quad (4)$$

где $\rho^{i+1,b}$ и $\rho^{i+1,p}$ – массы «рожденных» и «выживших» частиц соответственно, а p_B – задаваемая вероятность рождения частицы.

Обновление весов «выживших» частиц. В общем случае частицы ассоциируются с ячейкой лишь с определенной вероятностью, что моделирует ошибку сенсора. Тогда вес обновленных частиц ω^{i+1} будет выражаться формулой (5):

$$\omega^{i+1} = g^{i+1}(z^{i+1}) \cdot \omega^{i,p}, \quad (5)$$

где $g^{i+1}(z^{i+1})$ – функция правдоподобия измерения сенсора от существования в ячейке частицы. После этого веса в каждой ячейке нормализуются.

Инициализация новых частиц. Затем производится засевание новых частиц. При этом для того, чтобы реализация алгоритма работала в реальном времени, как количество новых частиц на каждой итерации, так и общее количество частиц постоянно. Веса «рожденных» частиц рассчитываются по формуле (6):

$$\omega^{i+1,B} = \frac{p_A^{i+1} \cdot \rho^{i+1,b}}{K_A}, \quad (6)$$

где p_A^{i+1} – вероятность ассоциирования новых частиц (в простейшем случае задается константой), а K_A – количество таких частиц в ячейке.

Вычисление среднего и дисперсии. Для предсказания положения частиц, а также для классификации ячеек вычисляется среднее, дисперсия и ковариация скоростей частиц (7):

$$\begin{aligned}\bar{v}_x &\approx \frac{1}{\rho^{i+1,p}} \sum_{k=1}^{K_p} \omega_k^{i+1} \cdot v_{x,k}^{i+1}, \\ \sigma_{v_x}^2 &\approx \frac{1}{\rho^{i+1,p}} \sum_{k=1}^{K_p} \omega_k^{i+1} \cdot (v_{x,k}^{i+1})^2 - \bar{v}_x^2, \\ cov(v_x, v_y) &\approx \frac{1}{\rho^{i+1,p}} \sum_{k=1}^{K_p} \omega_k^{i+1} \cdot v_{x,k}^{i+1} \cdot v_{y,k}^{i+1} - \bar{v}_x \cdot \bar{v}_y.\end{aligned}\quad (7)$$

Аналогичные формулы справедливы и для y компонент векторов скоростей частиц.

Перевыборка частиц. Отсевание как предсказанных с предыдущей, так и сгенерированных на текущей итерации частиц производится полиномиальным методом [19]. При этом чем больше вес частицы, тем больше у нее вероятность быть выбранной для следующей итерации алгоритма, а так как отсеивание имеет стохастический характер, то каждая ячейка может быть выбрана как 0, так и N раз, где N – общее количество частиц, гиперпараметр алгоритма.

Выделение динамики. Затем для разделения динамических и статических ячеек карты с пороговым значением τ_m сравнивается расстояние Махаланобиса (8):

$$m = \vec{v}_{(x,y)} \begin{pmatrix} \sigma_{v_x}^2 & cov(v_x, v_y) \\ cov(v_x, v_y) & \sigma_{v_y}^2 \end{pmatrix}^{-1} \vec{v}_{(x,y)}^T. \quad (8)$$

Также отметим, что вычисления по формулам (1–7) могут вычисляться по массивам частиц и ячеек параллельно и независимо друг от друга.

Метод [16] обладает рядом недостатков: во-первых, не учитывается собственное движение РТП между итерациями по построению карты. Во-вторых, максимальный модуль скорости частицы V_{max} при засеивании – гиперпараметр алгоритма. Слишком низкое значение ограничит возможность детектировать объекты с высокой скоростью, а высокое – увеличит количество ложных детекций (см. рис. 1). Наконец, при движении в реальных условиях на моментальной карте неизбежно будут возникать окклюдированные зоны, вызванные перекрытием поля зрения сенсора. Ячейки этих зон моментальной карты не будут засеиваться частицами, а значит будут создавать эффект движения на препятствиях с большими габаритами (см. рис. 2).

Компенсация собственного движения. Для учета собственного движения перед каждой итерацией фильтра выполняется смещение частиц в сторону, противоположную движению карты, при этом у всех частиц, вышедших за пределы карты, обнуляется вес, и они отсеиваются на этой итерации. Как правило, позицию и ориентацию платформы можно получить от инерциальной навигационной системы, однако в этом случае необходимо учесть асинхронность между получением кадров от датчиков системы технического зрения и данных одометрии, поскольку точность фильтра частиц сильно зависит от точности компенсации движения. В случае неправильного смещения, что особенно заметно на картах с небольшим размером дискрета, частицы динамических препятствий могут попадать на свободные или ячейки со статическими препятствиями, что в свою очередь приведет к потере движущегося объекта.

Фильтрация ложных детекций. Ошибочная классификация динамических ячеек характеризуется двумя свойствами: во-первых, неправильный расчет скорости происходит из-за превосходящего суммарного веса частиц, «рожденных» на текущей итерации, которые еще не успели отфильтроваться. Во-вторых, подобные выбросы одиночны. Поэтому дополнительно к сравнению с (8) те ячейки, суммарная масса «рожденных» частиц которых больше задаваемого порога и у которых в соседних ячейках нет динамических препятствий, считаются выбросами и отбрашиваются. Результат применения такого фильтра показан на рис. 1.

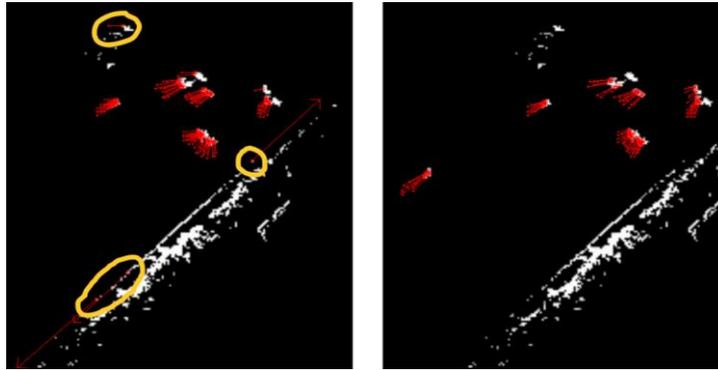


Рис. 1. Ложная классификация ячеек (скорости показаны красным) протяженного препятствия (слева) и результат фильтрации (справа)

Для уменьшения количества ложных детекций в окклюдированных зонах на вход фильтру частиц подается моментальная карта, дополненная ячейками статических препятствий накапливаемой карты занятости с предыдущей итерации алгоритма. Карта занятости по одному и тому же кадру до и после применения такой эвристики показана на рис. 2.

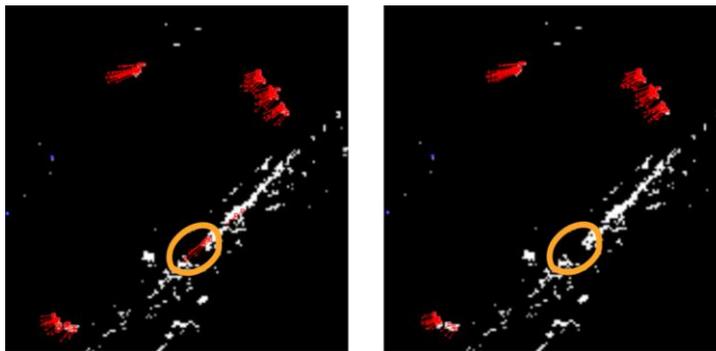


Рис. 2. Ложные детекции в окклюдированной зоне без объединения с накапливаемой картой (слева) и результат объединения (справа)

Наконец, облако точек, а следовательно и проекции на карту реальных динамических объектов в процессе движения не остаются статичными, что наиболее сильно сказывается на картах с высоким разрешением. Это связано как с изменением геометрии объектов (движение ног и рук пешеходов), так и с изменением точки обзора на объекты с постоянной геометрией. Все подобные изменения проекции динамических объектов будут помечаться как статические препятствия, поскольку в них не будут попадать частицы, сохранившиеся с предыдущих итераций. Для устранения этого эффекта использовался алгоритм связных компонент. На моментальной карте с его помощью находились группы ячеек-препятствий, затем, если среди этой группы обнаруживались динамические ячейки, к ним применялась операция дилатации до тех пор, пока не будут помечены все ячейки группы, либо радиус расширения не достигнет максимально возможного. На рис. 3 показана накапливаемая карта препятствий, полученная по алгоритму [17], и результат работы предлагаемого подхода по обнаружению динамических препятствий.

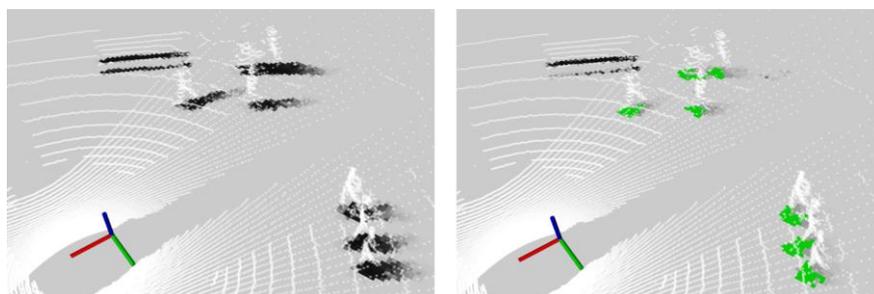


Рис. 3. Пешеходы на накапливаемой карте занятости без (слева) и с использованием (справа) фильтра частиц. Черным показаны статические препятствия, зеленым – динамические, серым – свободные ячейки

Поскольку фильтр частиц принимает на вход моментальные карты занятости и объединен с алгоритмом построения накапливаемой карты, то описанный подход к обнаружению динамических препятствий инвариантен к типу используемых сенсоров.

Экспериментальные результаты. Алгоритм был реализован на ядрах графического процессора с использованием языка CUDA. Быстродействие этой реализации замерялось на платформе Jetson AGX Xavier (запущенной в режиме энергопотребления MAXN) и на десктопном компьютере с процессором Intel i7-9700KF и видеокартой RTX 3070 на открытом наборе данных Waymo [20]. Также быстродействие программной реализации сравнивалось с подходами, описанными в работах [6, 7]. Результаты вычислительных экспериментов приведены в табл. 1.

Таблица 1

Быстродействие компонент алгоритма

	Подход [6]		Подход [7]		Предложенный подход	
	Desktop	Xavier	Desktop	Xavier	Desktop	Xavier
Детекция [6], мс	37	108	-	-	-	-
Сегментация [7], мс	-	-	39	77	-	-
Фильтр частиц, мс	-	-	-	-	20	30
Карты занятости, мс	15	20	33	64	21	57
Суммарное время, мс	52	128	72	141	41	87

Кроме того, описанные алгоритмы были протестированы в реальных условиях на сервисных РТП: уборщике и курьере (см. рис 4). На обеих платформах использовались лидары Ouster, инерциальные спутниковые системы Xsens MTi-680G и вычислительные блоки на базе Nvidia Jetson AGX Xavier. В табл. 2 указаны гиперпараметры фильтра частиц, использованные во всех проведенных экспериментах.

Таблица 2

Экспериментальные параметры алгоритма

Параметр	Значение	Параметр	Значение
N	80000	p_B	0,02
p_S	0,98	τ_m	6,0
p_A	1	V_{max} , м/с	20



Рис. 4. Робот-уборщик (слева) и робот-курьер (справа)

Выводы и будущая работа. В статье представлен улучшенный алгоритм построения карты занятости, в котором благодаря фильтру частиц происходит обнаружение динамических препятствий. Предложены несколько эвристик, уменьшающие количество ложных срабатываний и шумов и позволяющие использовать меньшее количество частиц и, следовательно, вычислительного времени, чем в оригинальной статье [16]. Разработанные модификации улучшают качество работы фильтра частиц и протестированы в реальных условиях и в среде с активным движением динамических объектов, что критически важно для сервисных роботов. Программная реализация метода обладает более высоким быстродействием, чем другие рассмотренные подходы к выделению динамики, и способна работать в реальном масштабе времени на встраиваемых платформах.

Существует несколько направлений дальнейшего развития описанного метода. Во-первых, необходимо разработать алгоритм комплексирования и синхронизации информации от моментальных карт с учетом используемого сенсора и временной асинхронности данных. Так облака точек от радара содержат информацию об относительной скорости объектов, которую можно использовать для более точной инициализации ячеек, а карты по данным стереокамеры могут содержать семантическую информацию о классе объекта, что также можно использовать при инициализации частиц. Во-вторых, алгоритм связных компонент, который объединяет отдельные ячейки в объекты, можно использовать для уточнения и предсказания траектории препятствий.

Благодарность. Экспериментальные исследования, в том числе оснащение робототехнической платформы и запись наборов данных, проведены совместно с ООО «ИнтеграНТ», г. Москва.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Elfes A.* Using occupancy grids for mobile robot perception and navigation // Computer. – 1989. – Vol. 22, No. 6. – P. 46-57.
2. *Geiger A., Lenz P., Urtasun R.* Are we ready for autonomous driving? the kitti vision benchmark suite // 2012 IEEE Conference on Computer Vision and Pattern Recognition. – IEEE, 2012. – P. 3354-3361.
3. *Caesar H. et al.* nuscenes: A multimodal dataset for autonomous driving // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2020. – P. 11621-11631.
4. *Shi S., Wang X., Li H.* pointcnn: 3d object proposal generation and detection from point cloud // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2019. – P. 770-779.
5. *Yin T., Zhou X., Krahenbuhl P.* Center-based 3d object detection and tracking // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2021. – P. 11784-11793.
6. *Шепель И.О.* Алгоритм комплексирования нескольких источников данных в общую карту занятости // Известия ЮФУ. Технические науки. – 2021. – № 3 (220). – С. 64-71.
7. *Shepel I. et al.* Occupancy Grid Generation With Dynamic Obstacle Segmentation in Stereo Images // IEEE Transactions on Intelligent Transportation Systems. – 2021.

8. Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – P. 3431-3440.
9. Eppenberger T. et al. Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking // 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – IEEE, 2020. – P. 10528-10535.
10. Cortinhal T., Tzelepis G., Erdal Aksoy E. SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds // International Symposium on Visual Computing. – Springer, Cham, 2020. – P. 207-222.
11. Xu J. et al. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation // Proceedings of the IEEE/CVF International Conference on Computer Vision. – 2021. – P. 16024-16033.
12. Mohajerin N., Rohani M. Multi-step prediction of occupancy grid maps with recurrent neural networks // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – P. 10600-10608.
13. Schreiber M., Hoermann S., Dietmayer K. Long-term occupancy grid prediction using recurrent neural networks // 2019 International Conference on Robotics and Automation (ICRA). – IEEE, 2019. – P. 9299-9305.
14. Danescu R., Nedeveschi S. A Particle-Based Solution for Modeling and Tracking Dynamic Digital Elevation Maps // IEEE Transactions on Intelligent Transportation Systems. – 2014. – Vol. 3, No. 15. – P. 1002-1015.
15. Tanzmeister G. et al. Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation // 2014 IEEE International Conference on Robotics and Automation (ICRA). – IEEE, 2014. – P. 6090-6095.
16. Nuss D. et al. A random finite set approach for dynamic occupancy grid maps with real-time application // The International Journal of Robotics Research. – 2018. – Vol. 37, No. 8. – P. 841-866.
17. Шенель И.О. Модифицированный алгоритм построения карты занятости по облаку точек от нескольких лидаров // Программные продукты и системы. – 2020. – Т. 33, № 2.
18. Dempster A.P. Upper and lower probabilities induced by a multivalued mapping // Classic works of the Dempster-Shafer theory of belief functions. – Springer, Berlin, Heidelberg, 2008. – P. 57-72.
19. Li T., Bolic M., Djuric P.M. Resampling methods for particle filtering: classification, implementation, and strategies // IEEE Signal processing magazine. – 2015. – Vol. 32, No. 3. – P. 70-86.
20. Sun P. et al. Scalability in perception for autonomous driving: Waymo open dataset // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2020. – P. 2446-2454.

REFERENCES

1. Elfes A. Using occupancy grids for mobile robot perception and navigation, *Computer*, 1989, Vol. 22, No. 6, pp. 46-57.
2. Geiger A., Lenz P., Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite, *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354-3361.
3. Caesar H. et al. nuscenes: A multimodal dataset for autonomous driving, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11621-11631.
4. Shi S., Wang X., Li H. pointcnn: 3d object proposal generation and detection from point cloud, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770-779.
5. Yin T., Zhou X., Krahenbuhl P. Center-based 3d object detection and tracking, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11784-11793.
6. Shepel' I.O. Algorithm kompleksirovaniya neskol'kikh istochnikov dannykh v obshchuyu kartu zanyatosti [Algorithm for combining multiple data sources into a common employment map], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2021, No. 3 (220), pp. 64-71.
7. Shepel I. et al. Occupancy Grid Generation With Dynamic Obstacle Segmentation in Stereo Images, *IEEE Transactions on Intelligent Transportation Systems*, 2021.

8. Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.
9. Eppenberger T. et al. Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10528-10535.
10. Cortinhal T., Tzelepis G., Erdal Aksoy E. SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds, *International Symposium on Visual Computing*. Springer, Cham, 2020, pp. 207-222.
11. Xu J. et al. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16024-16033.
12. Mohajerin N., Rohani M. Multi-step prediction of occupancy grid maps with recurrent neural networks, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10600-10608.
13. Schreiber M., Hoermann S., Dietmayer K. Long-term occupancy grid prediction using recurrent neural networks, *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9299-9305.
14. Danescu R., Nedevschi S. A Particle-Based Solution for Modeling and Tracking Dynamic Digital Elevation Maps, *IEEE Transactions on Intelligent Transportation Systems*, 2014, Vol. 3, No. 15, pp. 1002-1015.
15. Tanzmeister G. et al. Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation, *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6090-6095.
16. Nuss D. et al. A random finite set approach for dynamic occupancy grid maps with real-time application, *The International Journal of Robotics Research*, 2018, Vol. 37, No. 8, pp. 841-866.
17. Shepel' I.O. Modifitsirovannyi algoritm postroeniya karty zanyatosti po oblaku tochek ot neskol'kikh lidarov [A modified algorithm for constructing an employment map based on a cloud of points from several lidars], *Programmnye produkty i sistemy* [Software products and Systems], 2020, Vol. 33, No. 2.
18. Dempster A.P. Upper and lower probabilities induced by a multivalued mapping, *Classic works of the Dempster-Shafer theory of belief functions*. Springer, Berlin, Heidelberg, 2008, pp. 57-72.
19. Li T., Bolic M., Djuric P.M. Resampling methods for particle filtering: classification, implementation, and strategies, *IEEE Signal processing magazine*, 2015, Vol. 32, No. 3, pp. 70-86.
20. Sun P. et al. Scalability in perception for autonomous driving: Waymo open dataset, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446-2454.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор Г.В. Куповых.

Шепель Илья Олегович – Южный федеральный университет; e-mail: chepsiilya@gmail.com; г. Москва, Россия; тел.: +79085180350; кафедра математического обеспечения и применения ЭВМ; аспирант.

Shepel Ilya Olegovich – Southern Federal University; e-mail: chepsiilya@gmail.com; Moscow, Russia; phone: +79085180350; the department of mathematical support and computer applications; post-graduate.