ОБНАРУЖЕНИЕ И ОТСЛЕЖИВАНИЕ ПРЕПЯТСТВИЙ МОБИЛЬНОГО РОБОТА НА ОСНОВЕ ЛИДАРА

OBSTACLE DETECTION AND TRACKING BY MOBILE ROBOT BASED ON LADAR

Лю Вэй, магистрант, МГТУ им. Н. Э. Баумана, Москва, Российская Федерация

Liu Wei, lw1937478532@gmail.com

Аннотация

В настоящее время при исследовании лидара для обнаружения и отслеживания за подвижными препятствиями существуют проблемы: недостаточность кластеризации, ошибки при ассоциации препятствий в разных кадрах. Для решения этих проблем мы используем списки сеток для повышения полного спектра алгоритма кластеризации, а также возьмём несколько параметров препятствий для построения функций характеристик для повышения точности ассоциации препятствий. Во-первых, роботу необходимо выполнить растрирование данных облака точек, сегментировать слой земли, а затем использовать алгоритм DBSCAN для кластеризации облака точек. Во-вторых, робот использует несколько характеристик препятствия для ассоциации препятствия в разных кадрах, и, наконец, робот использует фильтр Калмана для отслеживания препятствия.

Annotation

At present, there are problems such as clustering insufficiency, obstacle association errors in different frames in lidar research for detection and tracking of moving obstacles. To solve these problems, we use grid lists to improve the sufficiency of clustering algorithm and use several obstacle feature functions to construct feature

functions to improve the accuracy of obstacle association. First, the Robot needs to rasterize the point cloud data, segment the ground layer, and then use the DBSCAN algorithm to cluster the point cloud. Second, the robot uses several obstacle characteristics to associate an obstacle in different frames, and finally, the robot uses the Kalman filter to track the obstacle.

Ключевые слова: Лидар, Облако точек, DBSCAN, Ассоциация препятствий, Калман.

Keywords: Lidar, Point cloud, DBSCAN, obstacle association, Kalman.

Мобильный робот, в условиях известной карты города, перемещается по дороге. Во время движения робот сталкивается с различными препятствиями. Текущие исследования по обнаружению и отслеживанию препятствий роботами в основном включают методы: на основе камеры [1] и на основе лидара [2]. Методы на основе камеры чувствительны к свету в окружающей среде, а лидар стал основным датчиком в обнаружении и отслеживании препятствий, благодаря своей высокой точности и низкому воздействию на окружающую среду.

В следующем разделе будет подробно описано, как робот может использовать лидар для обнаружения и отслеживания препятствий.

Создание карты сетки.

Робот использует лидар «Velodyne HDL 64E» для получения информации об окружающей среде. Информация, выводимая лидаром, включает информацию о положении препятствия в системе координат лидара и информацию об отраженной интенсивности лазерного луча. Исходные данные лидара преобразуются из системы координат лидара: $O - xy$ в систему координат изображения $O' - uv$, путем построения модели преобразования координат по следующему уравнению:

$$\begin{cases} x = \left(u - \frac{m}{2}\right)\frac{l}{m} \\ y = \left(\frac{n}{2} - v\right)\frac{w}{n} \end{cases}$$

Где: $m * n$ - разрешение карты сетки, а $l * w$ - прямоугольную область с центром на роботе. Мы выбираем $m = 600$ пикселей, $n = 400$ пикселей, $l = 60$ м, $w = 40$ м. Учитывая, что робот движется по городским дорогам, размер каждой сетки установлен равным 15 см $* 15$ см. Карта сетки показана на рисунке:



Рис.1. Исходное облако точек

Сегментация облака точек земли.

Сегментация облака точек земли является эффективным средством для повышения эффективности вычислений. После проецирования данных облака точек на двумерную сеточную карту, каждое облако точек P_i по-прежнему сохраняет значения координат: x, y, z и интенсивность отражения луча In_{P_i}, а именно $P_i = \{x_{P_i}, y_{P_i}, z_{P_i}, In_{P_i}\}$, каждая сетка G_i сохраняет координаты вершины, номер сетки и прочую информацию. Максимальные и минимальные высоты всех данных облака точек [3], проецируемых на одну и ту же сетку, составляют:

$$\begin{cases} h_{jmax} = \max\{ z_{P_i} \mid z_{P_i} \in G_j \} \\ h_{jmin} = \min\{ z_{P_i} \mid z_{P_i} \in G_j \} \end{cases}$$

Определите атрибуты сетки как:

$$f_{G_j} = h_{jmax} - h_{jmin}$$

Предполагая, что порог разности высоты $\varepsilon = 0.1m$, после прослеживания всех облаков точек удалите облако точек земли $f_{G_j} \leq \varepsilon$, оставив слой облака точек препятствий. Как показано：

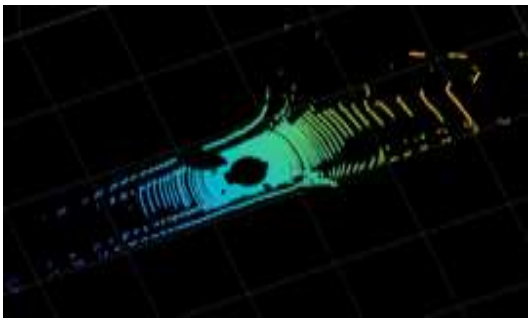


Рис.2. Облако точек земли

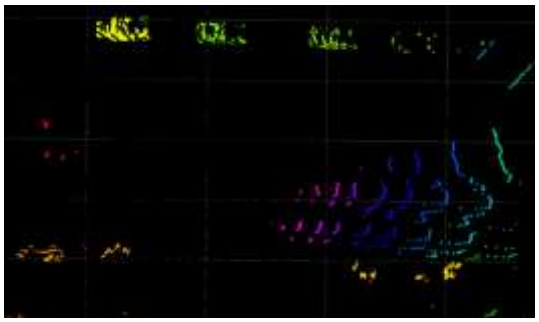


Рис.3. облака точек препятствий

Кластеризация облака точек препятствий.

Кластеризация облаков точек препятствий обычно использует метод кластеризации на основе плотности облака точек (DBSCAN) [4]. В алгоритме используется фиксированный параметр плотности облака точек. Алгоритм менее адаптивен, когда плотность облака точек препятствий на разных расстояниях от робота сильно различается. На основе удаления облака наземных точек мы создаем список сеток, чтобы помочь алгоритму DBSCAN в кластеризации：

(1) Создадим $GridList$ для хранения сеток препятствий и $ObjectList$ для хранения препятствий.

(2) Проследим всю карту сетки и поместим все сетки препятствий в список $GridList$.

(3) Случайным образом возьмём сетку из списка $GridList$ и отметим ее как $Object_i(i = 0,1,2,\dots)$. Удалим его из списка $GridList$ и поместим в список $ObjectList$.

(4) Определим, что окружающие его восемь соседних сеток принадлежат списку $GridList$, если это так, то сетка также помечается как $Object_i$ и удаляется из списка $GridList$. Если нет, то переходим к следующему шагу.

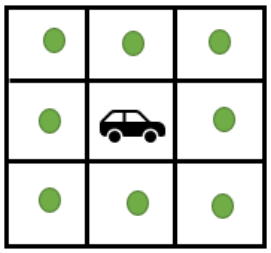


Рис.4. восемь соседних сеток

(5) Повторим шаги (3) и (4), пока список $GridList$ не станет пустым.

После завершения кластеризации препятствий, мы представляем препятствие внешним прямоугольником [5]. Параметры прямоугольника как: $s = (x_c, y_c, w, l, h, v, \phi)$, где (x_c, y_c) – координаты центра внешнего прямоугольника препятствия, w, l, h - ширина, длина и высота препятствия, соответственно. v-скорость центральной точки препятствия, а ϕ - угол между направлением движения препятствия и осью y системы координат лидара.

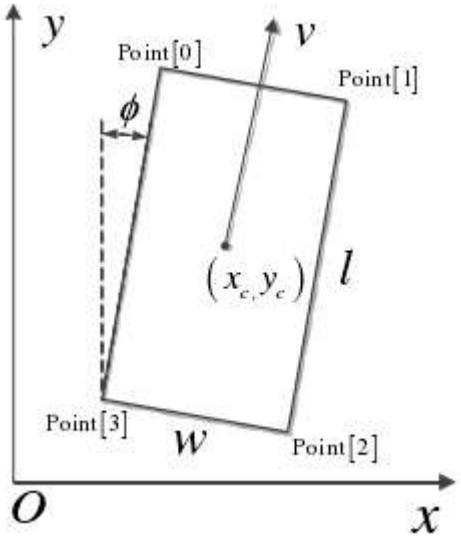


Рис.5. Внешний прямоугольник препятствия

Результаты сравнительного эксперимента:

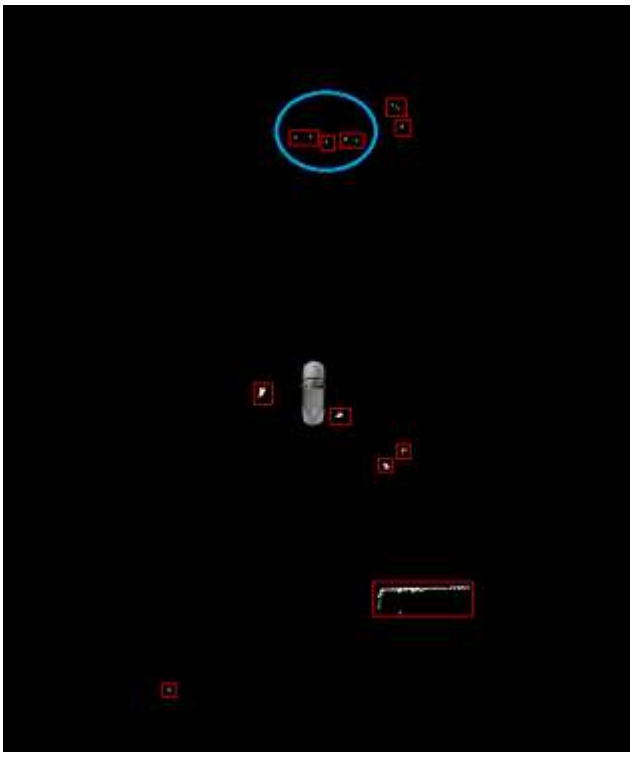


Рис.6. Традиционный DBSCAN

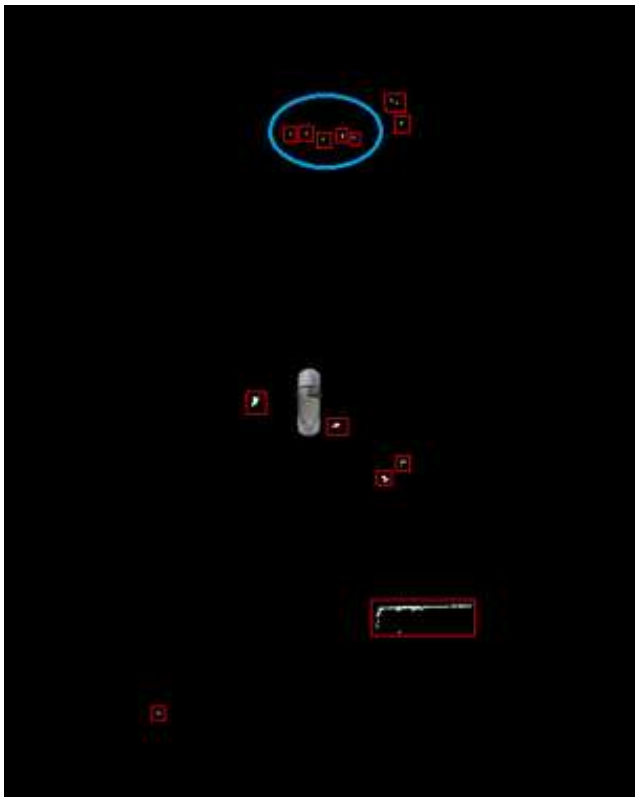


Рис.7. Оптимизированный DBSCAN

Алгоритм DBSCAN более эффективен в кластеризации плотных облаков точек на близких расстояниях, а когда препятствия находятся далеко от робота, облака точек разрежены, в результате чего пять удаленных пешеходов формируются в три кластера из-за использования фиксированных параметров, как на рис. 6, в то время как оптимизированный алгоритм может точно формироваться в кластер препятствия.

Ассоциация препятствий.

Период сканирования лидара короткий, и данные соответствующие одному и тому же углу сканирования между соседними кадрами, имеют большую ассоциацию [6]. Ассоциация препятствий заключается в том, чтобы сопоставить препятствие, наблюдаемое в текущий момент, с целью с наименьшей разницей в списке препятствий, установленной в предыдущий момент. Функция определения местоположения препятствия обычно используется для оценки, но она чувствительна к шумовым помехам. Для повышения точности ассоциации препятствий, мы используем положение, соотношение сторон, высоту и плотность облака точек. Чтобы установить функцию степени различия. После вычисления значений функций разности между препятствиями в текущем кадре и всеми препятствиями в предыдущем кадре, мы выбираем два препятствия с наименьшей разницей для ассоциации.

Предположим, что функция степени разности между $i-$м препятствием, обнаруженным в $T-$м кадре, и $j-$м препятствием, обнаруженным в $T+1-$м кадре, составляет:

$$D(i,j) = \alpha_1\delta_{\text{dist}} + \alpha_2\delta_{\text{ratio}} + \alpha_3\delta_{\text{height}} + \alpha_4\delta_{\text{density}}$$

Среди них:

Разница расстояний: $\delta_{\text{dist}} = \sqrt{\left(x^j_{\text{c},T+1} - x^i_{\text{c},T}\right)^2 + \left(y^j_{\text{c},T+1} - y^i_{\text{c},T}\right)^2}$

Разница в соотношении сторон: $\delta_{\text{ratio}} = \left(\frac{l^j_{T+1}}{w^j_{T+1}} - \frac{l^i_T}{w^i_T}\right)$;

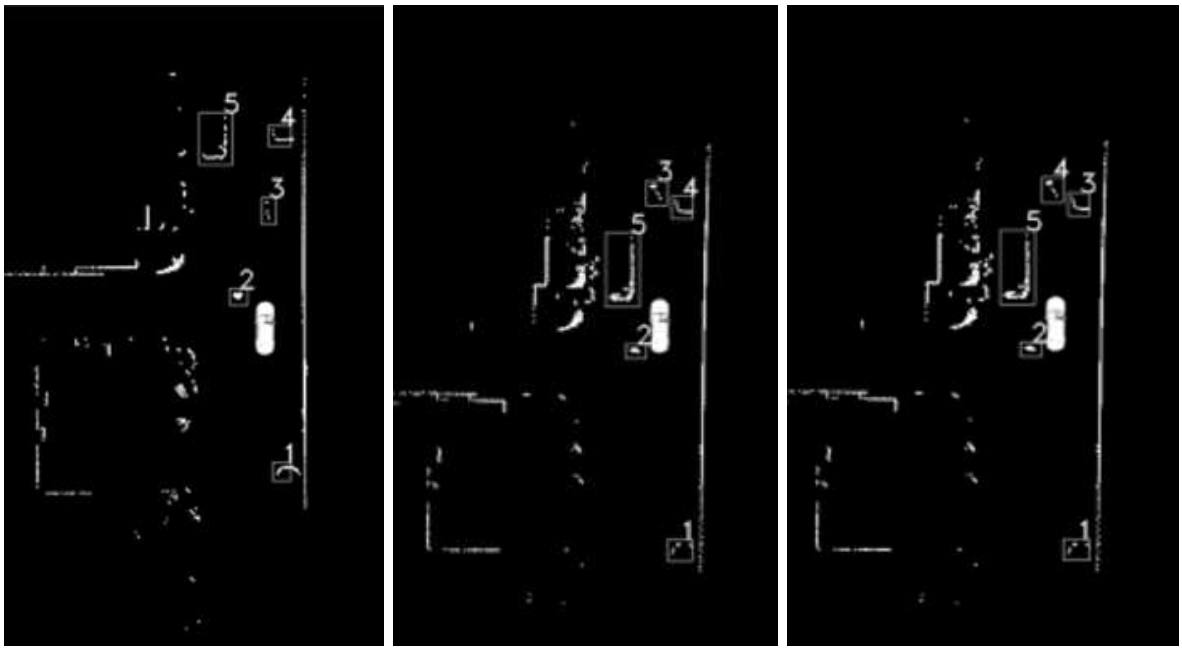
Разница высоты: $\delta_{\text{height}} = \left(h^j_{T+1} - h^i_T\right)$;

Разница плотности облака точек в сетке:

$$\delta_{\text{density}} = \frac{\sum\sum p\left(u_{T+1}^{j}, v_{T+1}^{j}\right)}{w_{T+1}^{j} l_{T+1}^{j}} - \frac{\sum\sum p\left(u_{T}^{i}, v_{T}^{i}\right)}{w_{T}^{i} l_{T}^{i}}$$

Среди них $p(u, v)$ используется, чтобы определять, есть ли облако точек в точке (u, v), если оно существует, то равно 1, а если его нет, то 0; $\alpha_1, \alpha_2, \alpha_3, \alpha_4$-четыре значения веса соответственно для настройки веса различных характеристик.

Результаты сравнительного эксперимента:



(a) (b) (c)

Рис.8. Ассоциация препятствий

На рис.8 (a) показан результат маркировки препятствия в кадре 15, а (b) и (c) - результаты оптимизированного и традиционного алгоритма ассоциации в кадре 25, соответственно; традиционный алгоритм неверно ассоциирует препятствие №3 в кадре 25 с препятствием №4 в кадре 15, а оптимизированный алгоритм может точно ассоциировать препятствия.

Отслеживание подвижных препятствий.

Мы используем фильтр Калмана для отслеживания препятствий. После ассоциаций препятствий мы вводим математические модели различных целевых

препятствий в фильтр Калмана и обновляем уравнение, чтобы объединить текущие данные наблюдения с предыдущими оценочными данными, а затем вывести их.

Уравнение предсказания Калмана:

$$x_k = Ax_{k-1}$$

Среди них переменные состояния $x = \left[x_{\text{с}}, y_{\text{с}}, v_{\text{с}x}, v_{\text{с}y}\right]^T$, $v_{\text{с}x}$ и $v_{\text{с}y}$ являются компонентами скорости препятствия v в направлениях x и y соответственно; A - матрица перехода состояний, а именно

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ T & 0 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$ Т – период

Матрица ковариации ошибки предсказания:

$$P_k = AP_{k-1}A' + \Gamma Q \Gamma'$$

Начальное значение P_k определяется, как единичная матрица четвертого порядка, а ковариационная матрица технологического шума и матрица распределения соответственно:

$Q = \sigma_{\text{p}}^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Gamma = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{\text{T}}$, σ_{p}-Технологический шум

Уравнение обновления наблюдений Калмана:

$$\begin{cases} K = P_{k-1}H^{\text{T}}\text{T}\, HP_{k-1}H^{\text{T}+R^{-1}} \\ x_k = x_{k-1} + K(z_k - H\, x_{k-1}) \\ P_k = P_{k-1} - KHP_{k-1} \end{cases}$$

Где: $z = [x_{\text{с}}, y_{\text{с}}]^T$, H - матрица наблюдения; R - ковариационная матрица шума измерения. Выражение матрицы H и R является:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, R = \sigma_{\text{r}}^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\sigma_{\text{r}} - \text{Шум измерения}$$

Погрешность измерения получается путем предварительного измерения относительно устойчивых препятствий, а затем получается ковариация шума измерения R. Ковариацию технологического шума Q обычно трудно получить

напрямую, и ее необходимо сначала определить, а затем отрегулировать в автономном режиме в соответствии с операцией программы [7]. Регулируя ковариацию технологического шума Q, результаты прогнозирования положения и скорости оцениваются препятствия в текущем кадре. Достоверность наблюдаемого значения регулируется путем настройки ковариации шума измерения R; склоняется ли окончательный выходной результат текущего кадра, к предсказанному значению или наблюдаемому значению, определяется параметрами дисперсии матрицы Q и R.

Результаты экспериментов.

Мы создали проект в программе «Visual Studio 2017» и обработали данные облака точек лидара с помощью библиотеки облаков точек PCL (Point Cloud Library), и проект содержал все вышеупомянутые алгоритмы. Мы выбрали сцену, в которой подвижное препятствие обходит робота с левой стороны.

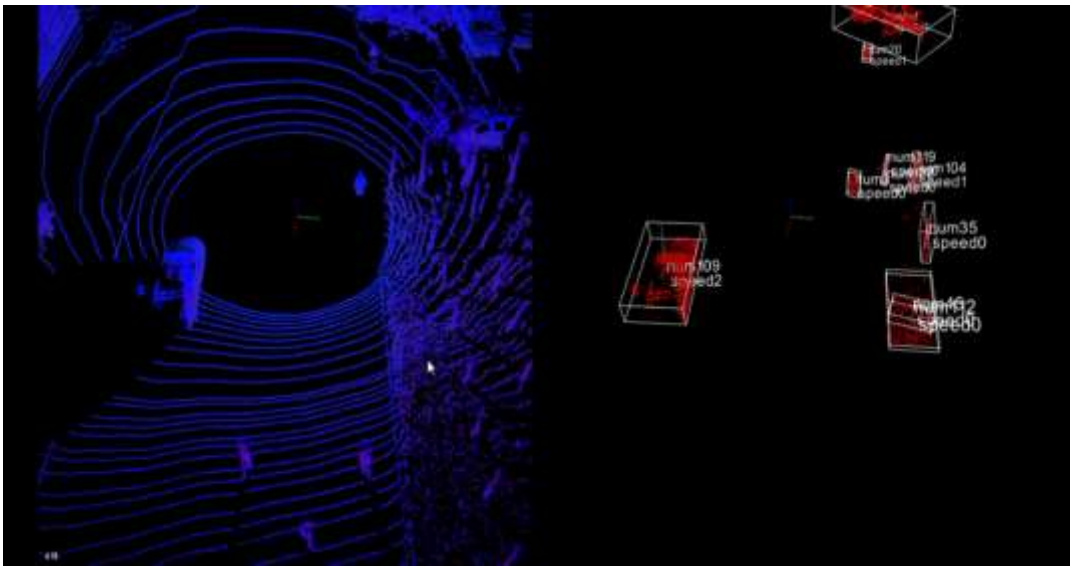


Рис.9. Кадр 418 Облако точек препятствий

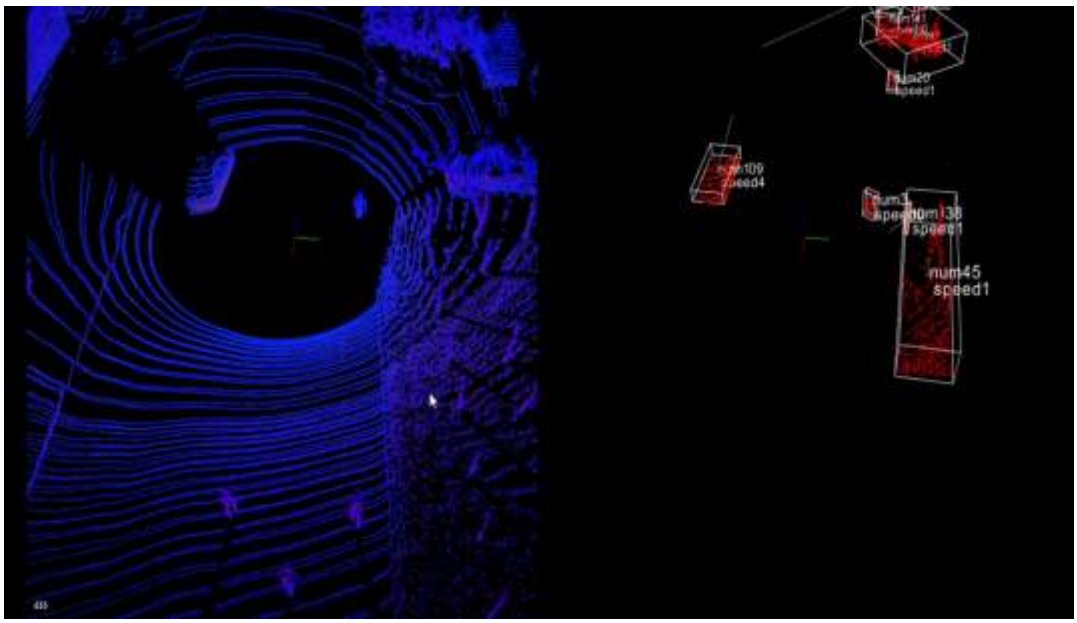


Рис.10. Кадр 433 Облако точек препятствий

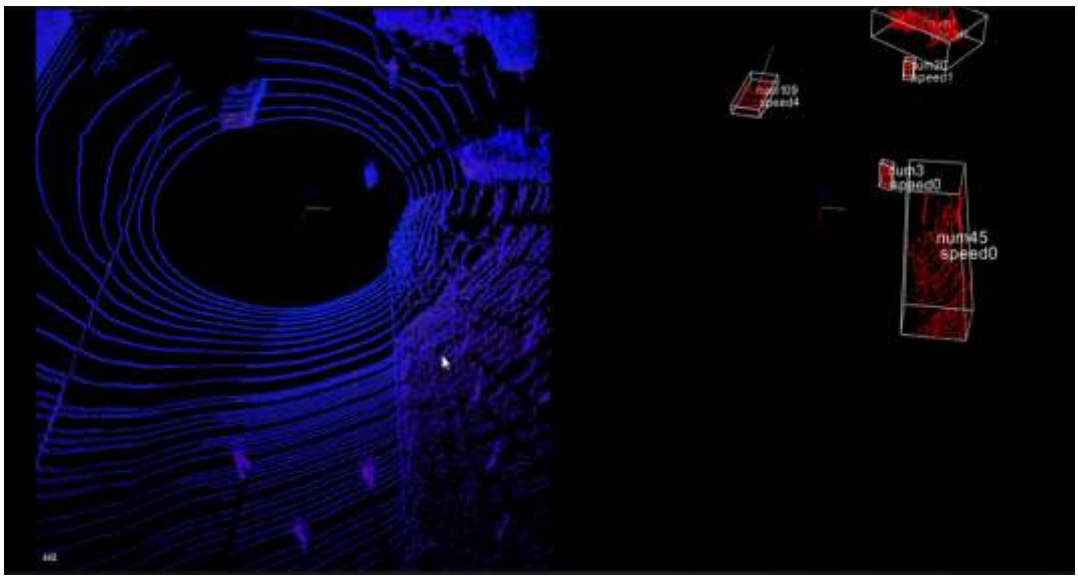


Рис.11. Кадр 442 Облако точек препятствий

Из результатов эксперимента видно, что метки различных препятствий в кадрах 418, 433 и 442 не изменились, и робот точно отслеживал подвижные препятствия без сбоев ассоциации и ошибок отслеживания, что доказало эффективность алгоритма в данной работе.

Вывод.

Эта статья в основном посвящена исследованию обнаружения и отслеживания подвижных препятствий на основе 3D-лидара. В процессе кластеризации препятствий мы предложили метод использования списков сеток

для помощи алгоритму кластеризации плотности (DBSCAN). Результаты экспериментов показывали, что оптимизированный алгоритм имеет более высокую точность кластеризации, так как использование одной характеристики привело бы к неправильной ассоциации препятствий. Мы использовали несколько характеристик препятствий для ассоциации, что повысило точность ассоциации препятствий, и наконец, использовали фильтр Калмана для отслеживания препятствий. Результаты экспериментов показали, что робот может точно отслеживать подвижные препятствия в зоне сканирования лидара.

Литература

1. ГАНДИ Т., ТРИВЕДИ М. М. Системы защиты пешеходов: проблемы, исследования и проблемы [J]. IEEE Transactions по интеллектуальным транспортным системам, 2007, 8 (3): 413-430.
2. VU T D, BURLET J, AYCARD O. Локализация на основе сетки и локальное отображение с обнаружением и отслеживанием движущихся объектов [J]. Информационное объединение, 2011, 12 (1): 58-69.
3. Синь Ю, Лян Хуахуа, Мэй Тао и др. Метод обнаружения и отображения динамических препятствий в беспилотных автомобилях на основе лазерных датчиков [J]. Робототехника, 2014, 36 (6): 654-661.
4. Су Чжиюань, Сюй Ючунь, Ли Юнлэ. Метод обнаружения транспортных средств на основе трехмерного лазерного радара [J]. Журнал Академии военного транспорта, 2017, 19 (1): 45-49.
5. Ван Сяо, Ли Кэцян, Ван Цзяньцян и др. Интеллектуальная идентификация параметров цели транспортного средства на основе трехмерного лазерного радара [J]. Автомобильная инженерия, 2016, 38 (9): 1146-1152.
6. Ван Шэн. Исследование методов обнаружения роботизированных препятствий в сложной полевой среде [D]. Ханчжоу: Чжэцзянский университет, 2015.
7. Чжоу Цзюньцзин, Дуань Цзяньминь, Ян Гуанцзу и др. Метод идентификации и слежения за автомобилем на основе.

Literature

1. GANDHI T, TRIVEDI M M. Pedestrian protection systems: issues, survey, and challenges[J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(3): 413-430
2. VU T D, BURLET J, AYCARD O. Grid-based localization and local mapping with moving object detection and tracking[J]. Information Fusion, 2011, 12(1): 58-69.
3. Xin Yu, Liang Huahua, Mei Tao, etc. Detection and representation method of dynamic obstacles in unmanned vehicles based on laser sensors [J]. Robotics, 2014, 36(6): 654-661.
4. Su Zhiyuan, Xu Youchun, Li Yongle. Vehicle target detection method based on three-dimensional laser radar [J]. Journal of the Academy of Military Transportation, 2017, 19(1): 45-49.
5. Wang Xiao, Li Keqiang, Wang Jianqiang, etc. Intelligent vehicle target parameter identification based on three-dimensional laser radar [J]. Automotive Engineering, 2016, 38(9): 1146-1152.
6. Wang Sheng. Research on the detection method of robot obstacles in complex field environment [D]. Hangzhou: Zhejiang University, 2015.
7. Zhou Junjing, Duan Jianmin, Yang Guangzu, etc. Vehicle identification and tracking method based on radar ranging [J]. Automotive Engineering, 2014 (11): 1415-1420.