Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Масштабируемые алгоритмы одновременного построения карты и локализации стаи мобильных роботов

Филатов Антон Юрьевич

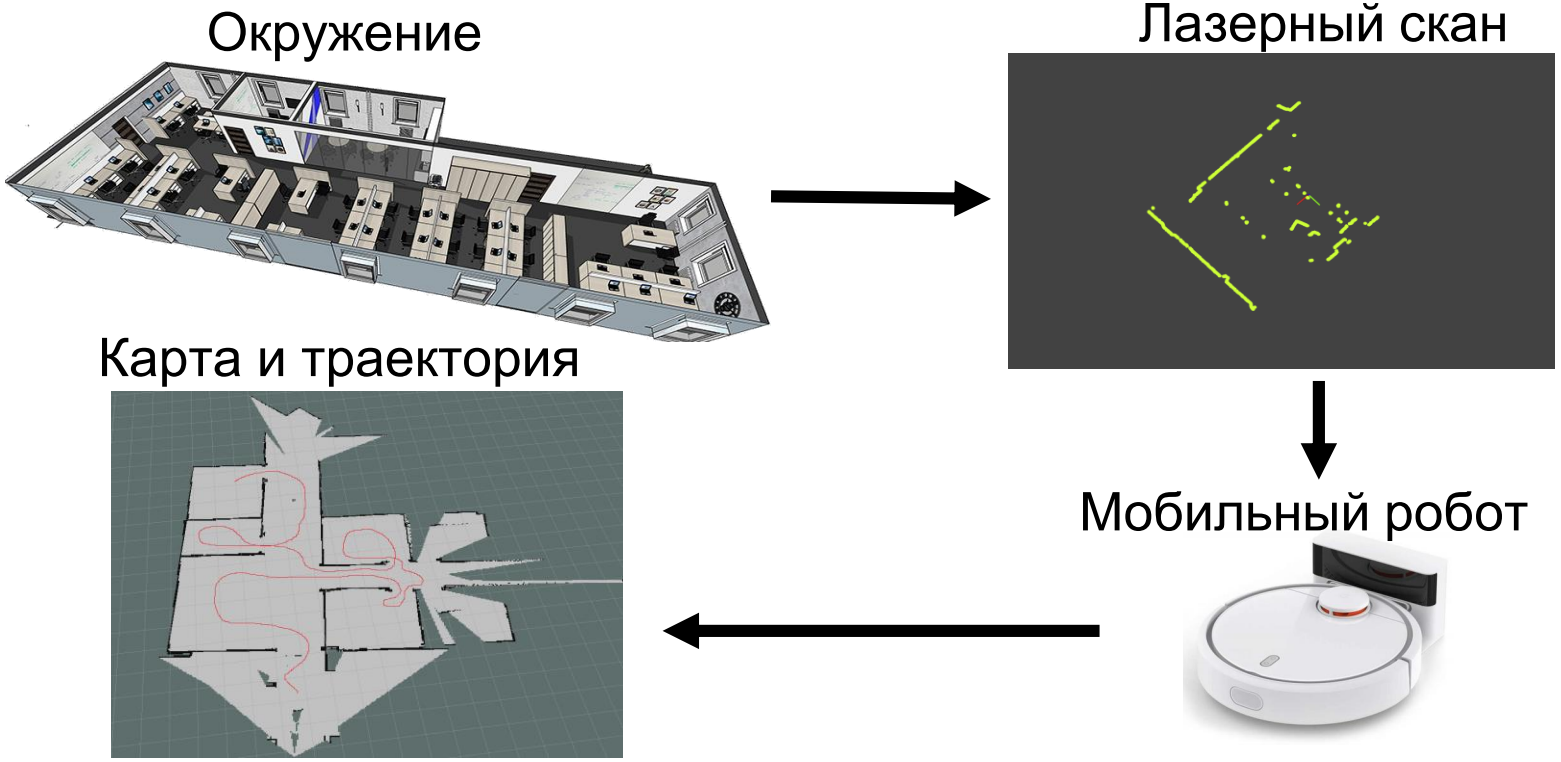
СПЕЦИАЛЬНОСТЬ: 05.13.11 “МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ МАШИН, КОМПЛЕКСОВ И КОМПЬЮТЕРНЫХ СЕТЕЙ”

Научный руководитель: кандидат технических наук, доцент
Кринкин Кирилл Владимирович

Санкт-Петербург, 2021

Актуальность темы исследования (1)

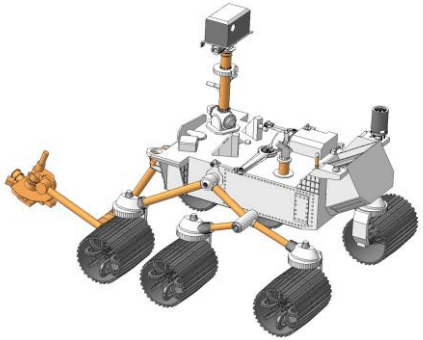
SLAM – одновременная локализация и построение карты
Мобильный робот – подвижная платформа + сенсоры + ЦПУ



Актуальность темы исследования (2)

Невозможно пользоваться: спутниковой навигацией, планом местности.
Опционально: непосредственное управление мобильным роботом.

- Применяется в задачах построения плана здания или местности в т.ч. для роботов-пылесосов, для проведения спасательных работ и для марсоходов.

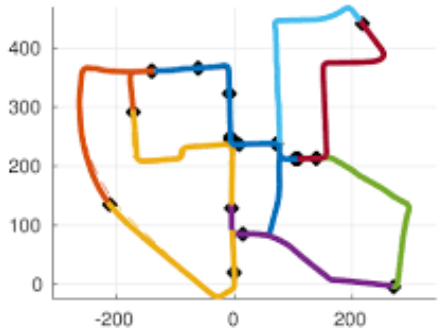


- Применяется в задачах автоматизации движении беспилотного транспорта (Яндекс, Uber, Toyota).

Актуальность темы исследования (3)

При решении любой из вышеперечисленных задач **уменьшение вычислительных затрат** на работу алгоритма SLAM позволяет перенаправить освободившиеся ресурсы на решение других востребованных задач.

Многоагентные алгоритмы позволяют **ускорить** процесс построения карты окружения за счёт распределения задач, а также **увеличить точность** построения за счёт взаимной корректировки.



Объект, предмет и цель исследования

Цель исследования:

Разработка масштабируемых алгоритмов для многоагентного решения задачи SLAM, применимых для низкопроизводительных роботов.

Объектом исследования является взаимодействие стаи мобильных роботов при решении задачи одновременного построения карты и локализации.

Предметом исследования являются архитектура, точность и быстродействие многоагентного алгоритма, решающего задачу одновременного определения положения и построения карты стаи мобильных роботов.

Задачи исследования

1. Исследование и классификация существующих одноагентных и многоагентных алгоритмов, решающих задачу SLAM.
2. Разработка архитектуры компонентнов масштабируемого многоагентного алгоритма SLAM на базе теории Демпстера-Шафера.
3. Разработка масштабируемого алгоритма фильтрации двумерных лазерных сканов для освобождения вычислительных ресурсов.
4. Программная реализация масштабируемого многоагентного алгоритма решения задачи SLAM.
5. Определение характеристик точности многоагентного алгоритма, а также границ применимости на низкопроизводительных роботах.

Существующие многоагентные алгоритмы

На основании работ Д. Скарамуззы (h: 65), К. Жоу (h: 11), М. Лазаро(h: 7) можно разделять многоагентные SLAM алгоритмы на следующие категории:

- Централизованные и **децентрализованные;**
- С известным начальным взаимным расположением агентов и **неизвестным;**
- С синхронной или **асинхронной** моделью обмена данными.

Выделенные категории не накладывают ограничений на окружающую среду и на техническое оснащение агентов.

Помимо этой классификации к многоагентным алгоритмам применима классификация одноагентных алгоритмов.

Классификация одноагентных алгоритмов

Предложенная Х. Дюррант-Уайтом (h: 82) и С. Труном (h: 154) классификация:

	Вид карты	Предобработка	Число одновременных гипотез о мире	Название алгоритма	Рейтинг	
					Точн.	Скор.
2D	Графовые	Сырые данные	одногипотезные	**Google Cartorgapher**	<0,3м	25 с/с
	Неграфовые	Сырые данные	одногипотезные	**vinySLAM**	<0,4м	50 с/с
			многогипотезные	**Gmapping**	<0,4м	30 с/с
		Особые точки	одногипотезные	EKF-SLAM	<1м	25 с/с
			многогипотезные	FastSLAM	<1,1м	60 с/с
3D	Графовые	Особые точки	одногипотезные	ORB-SLAM	< 6 м	20 к/с
	Неграфовые			VSLAM	< 1 м	25 к/с

Положения, выносимые на защиту

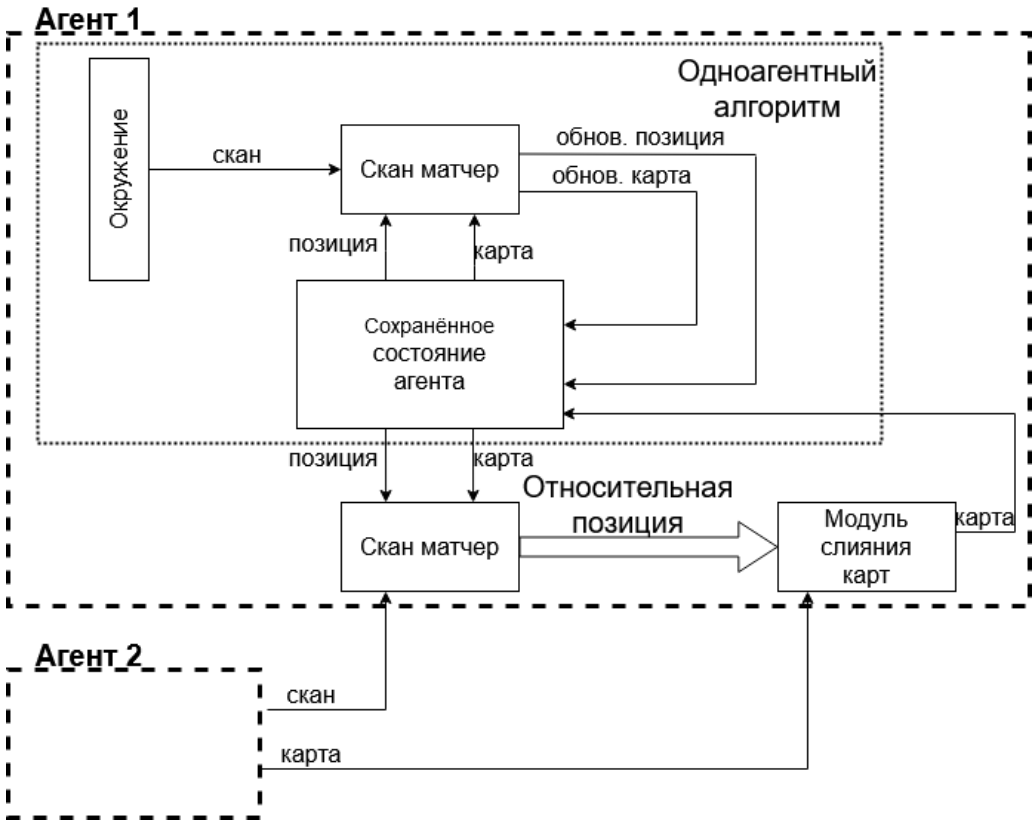
1. Масштабируемый многоагентный алгоритм, решающих задачу SLAM для стаи мобильных роботов, на базе теории Демпстера-Шафера.
2. Масштабируемый алгоритм фильтрации двумерных лазерных сканов для многоагентного алгоритма SLAM.
3. Программная реализация масштабируемого многоагентного алгоритма, решающего задачу SLAM для стаи мобильных роботов.

1. Масштабируемый многоагентный алгоритм, решающих задачу SLAM для стаи мобильных роботов, на базе теории Демпстера-Шафера

Ограничения для многоагентного алгоритма

- Все роботы оснащены одинаково и имеют равные права и обязанности.
- **Каждый робот выполняет алгоритм SLAM независимо от других.**
- Роботы обмениваются данными, когда один робот оказывается в окрестности другого робота.
- Вычисляется взаимное расположение роботов и обмен построенными ими картами.
- Объединение карт после определения взаимного расположение осуществляется с использованием теории Демпстера-Шафера.
- Горизонтальное масштабирование алгоритма достигается за счёт обеспечения автономной работы каждого агента до тех пор, пока два агента не встретятся.

Вычислительная среда алгоритма SLAM для стаи мобильных роботов[1]



Каждый робот выполняет независимо от других одноагентный SLAM алгоритм, пока какие-то роботы не встретятся.

1. Filatov, A., Krinkin, K. A Simplistic Approach for Lightweight Multi-Agent SLAM Algorithm. // International Journal of Embedded and Real-Time Communication Systems (IJERTCS). - 2020 - Т. 1. - № 3. - С. 67-83.

Метод определения взаимного расположения

```
def match (observation, pose, map) :
    for random position ∈ neighborhood(pose) do
        pose_score ← try_insert(position + observation, map)
    end for
    return position[best_score]
end def
```

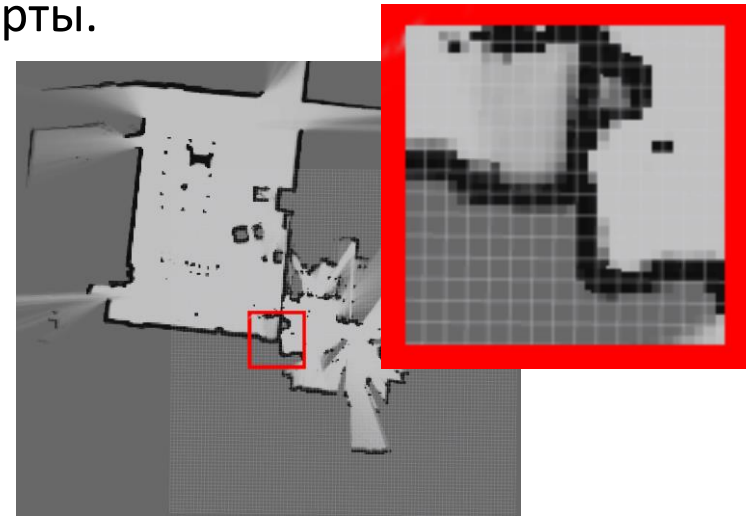
```
current_pose_i, map_i ← match(obs_i, current_pose_{i−1}, map_{i−1})
if another_robot_is_near() then
    foreign_map, foreign_observation ← receive_foreign_state()
    foreign_pose ← match(foreign_observation, current_pose_i, map_i)
end if
```

Для обеспечения быстродействия использован алгоритм стохастического поиска, показывающий достаточно точные результаты[2]

2. Krinkin K. The scan matchers research and comparison: Monte-carlo, olson and hough/ Krinkin K., Filatov A., Filatov A., Huletski A., Kartashov D. //2016 19th Conference of Open Innovations Association (FRUCT). – IEEE, 2016. – С. 99-105.

Теория Демпстера-Шафера

Применяется для описания модели ячейки карты.



Без ТДШ используется Байесовский подход:

1	0,9
0	0,5

Каждая ячейка содержит вероятность быть занятой.

$$p_{\cup} = \frac{1}{n}\sum_{i=1}^{n} p_i$$

По ТДШ:

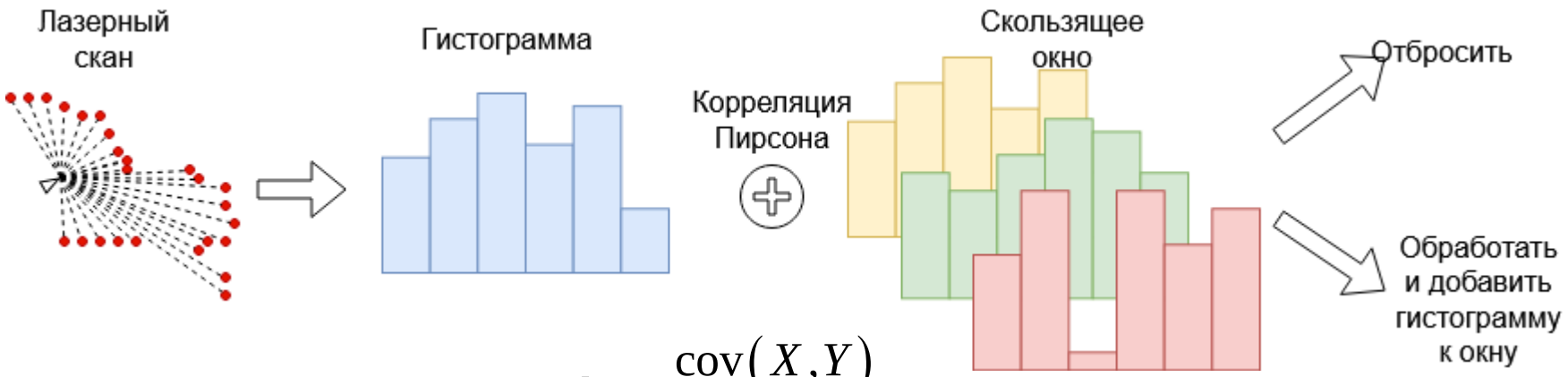
O	F	O	F
B	C	B	C
O	F	O	F
B	C	B	C

Каждая ячейка содержит 4 массы:
O – occupied – масса занятости,
F – free – масса свободы,
B – both – масса быть в обоих состояниях одновременно,
C – conflict – масса не быть ни в одном состоянии

$$m_{\cup}(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = 0} m_1(B) m_2(C)}$$

2. Масштабируемый алгоритм фильтрации двумерных лазерных сканов для многоагентного алгоритма SLAM

Схема алгоритма фильтрации[3]



Корреляция Пирсона: $k_P = \frac{\mathrm{cov}(X,Y)}{\sigma_X \cdot \sigma_Y}$

Корреляция Спирмана: $k_S = \frac{\mathrm{cov}(rgX, rgY)}{\sigma_{rgX} \cdot \sigma_{rgY}}$, где rg – ранг СВ (функция порядка)

Корреляция Кендалла: $k_K = 1 - \frac{4Q}{N(N-1)}$, где Q – количество несовпадающих рангов, меньших, чем значение ранга второй случайной величины

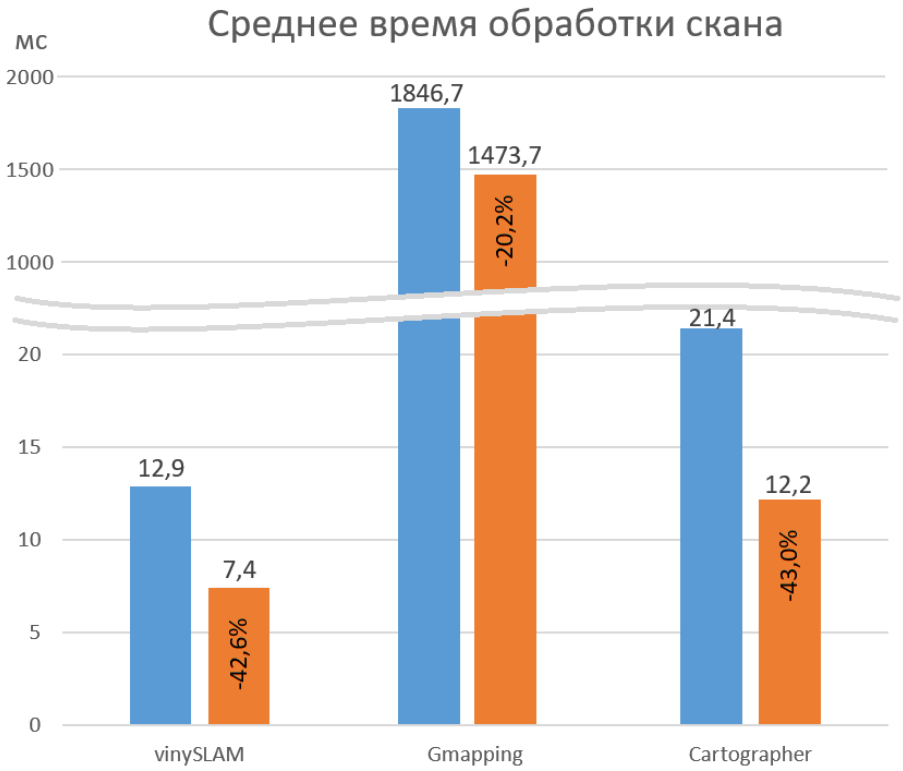
3. Krinkin K. The scan matchers research and comparison: Monte-carlo, olson and hough/ Krinkin K., Filatov A., Filatov A., Huletski A., Kartashov D. //2016 19th Conference of Open Innovations Association (FRUCT). – IEEE, 2016. – С. 99-105.

Описание работы предложенного алгоритма фильтрации

- Фильтр определяет, вносит ли новый лазерный скан новую информацию в систему, в противном случае не передаёт этот скан на обработку алгоритму SLAM.
- Время работы фильтра значительно меньше времени, затрачиваемого на работу скан матчера.
- Для того, чтобы избежать избыточного отбрасывания сканов, содержащих очень мало особых точек, разработан быстрый алгоритм определения прямых линий.
- Горизонтальное масштабирование алгоритма фильтрации достигается за счёт распределения построения гистограмм на несколько потоков.

- Проведённые эксперименты показали, что можно отбросить более 45% лазерных сканов.

Выигрыш времени при использовании фильтра



Для алгоритмов vinySLAM и Google Cartographer среднее время обработки сканов уменьшилось на 40%.

Выигрыш по времени для алгоритма Gmapping меньше из-за внутренней особенности этого алгоритма.

3. Программная реализация масштабируемого многоагентного алгоритма, решающего задачу SLAM для стаи мобильных роботов.

Псевдокод алгоритма многоагентного решения задачи SLAM[4]

while $True$ **do**
  $current\_pose_i, map_i \leftarrow \text{doSingleSlam}(obs_i, current\_pose_{i-1}, map_{i-1})$
  if another_robot_is_near() **then**
    $foreign\_map, foreign\_observation \leftarrow$ receive_foreign_state()
    $foreign\_pose \leftarrow \text{match}(current\_pose_i, map_i, foreign\_observation)$
    $map_i \leftarrow \text{combine\_with\_thDS}(map_i, foreign\_map)$
  end if
end while

doSingleSLAM – процесс выполнения «ядра» алгоритма. На этом месте может быть любой алгоритм, решающий задачу SLAM.

match – определить взаимное расположение агентов и получить координаты другого агента в собственной системе координат.

4. Филатов А.Ю., Кринкин К.В., Программа для вычисления взаимного расположения агентов, выполняющих алгоритм решения задачи SLAM. Номер свидетельства RU 2020610524 Дата публикации: 15.01.2020

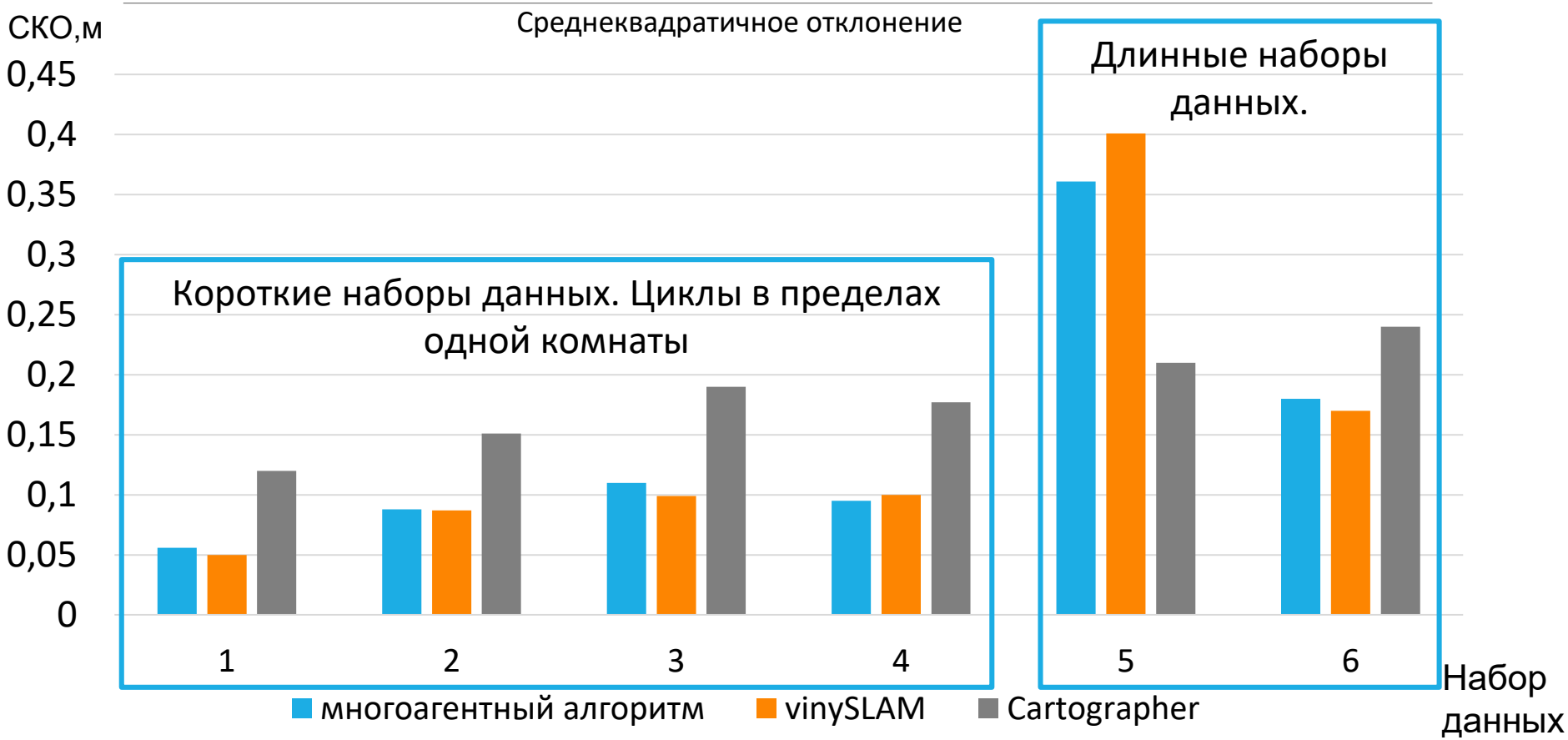
Описание эксперимента для измерения точности

1) Запустить разработанный алгоритм на записанных заранее наборах данных. Наборы данных записаны во время движения реального робота.
2) Запустить другие алгоритмы на тех же наборах данных.
3) Вычислить среднеквадратичное отклонение построенной каждым алгоритмом траектории движения от истинной траектории.

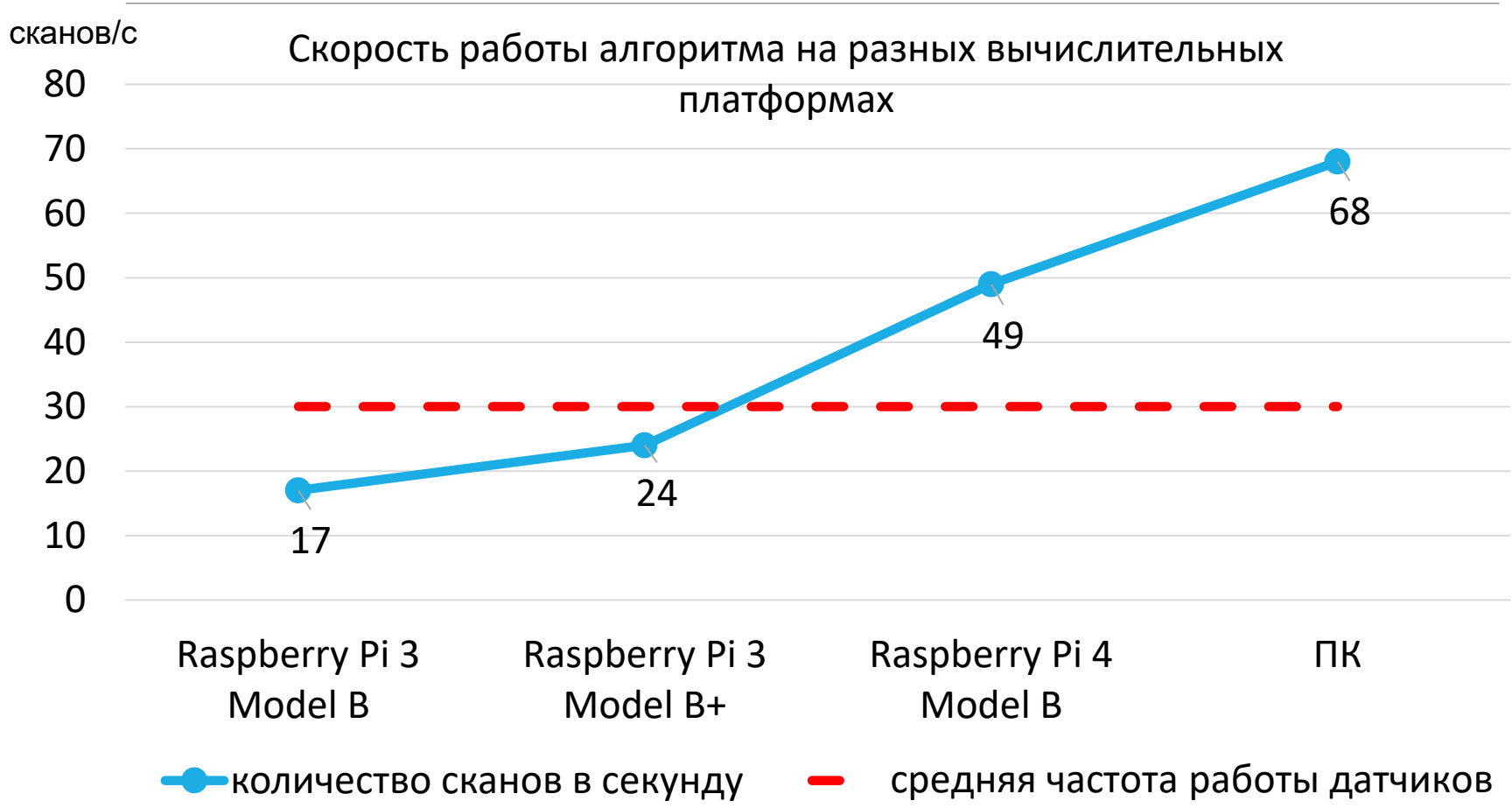
В качестве набора данных, содержащего истинную траекторию движения робота в помещении, а также содержащего 2D данные выбран **набор данных MIT.**

Среди рассматриваемых последовательностей данных присутствуют 4 последовательности с короткой траекторией движения с частыми самопересечениями траектории. А также 2 с длинной траекторией и редкими самопересечениями траектории

Количественные характеристики точности



Количественные характеристики скорости работы



Заключение

В ходе выполнения работы были решены следующие задачи:

1. Проклассифицированы алгоритмы, решающие задачу SLAM.
   - Предметом исследования стали стохастические неграфовые алгоритмы без использования фильтра частиц.
   - Разработана иерархия агентов, подходящая для масштабирования.
2. Предложена архитектура компонентов масштабируемого многоагентного алгоритма SLAM на базе теории Демпстера-Шафера.
   - Разработанный алгоритм имеет низкую вычислительную сложность за счёт применения стохастических методов, а также отказа от графовой структуры алгоритма и применения теории Демпстера-Шафера.
3. Разработан масштабируемый алгоритм фильтрации двумерных лазерных сканов.
   - Разработанный алгоритм фильтрации позволяет сократить время среднее обработки лазерного скана на 40%.
4. Разработано программное обеспечение, реализующее масштабируемый многоагентный алгоритм SLAM для стаи роботов.
5. Разработана методика тестирования многоагентного алгоритма, согласно которой точность работы алгоритма выше, чем алгоритм Google Cartographer, а также алгоритм обрабатывает от 24 сканов в секунду.

Апробация результатов

Выступления на конференциях:
FRUCT19 *Нояб 2016* The Scan Matchers Research and Comparison: Monte-Carlo, Olson and Hough;
FRUCT21 *Нояб 2017* 2D SLAM quality evaluation methods;
FRUCT22 *Май 2018* The Scan Matchers Research and Comparison: Monte-Carlo, Olson and Hough;
FRUCT22 *Май 2018* Evaluation of Modern Laser Based Indoor SLAM Algorithms;
FRUCT24 *Апр 2019* Multi-Agent SLAM Approaches for Low-Cost Platforms;
РЖД НИИАС семинар *Сен 2019* Применение теории Демпстера-Шафера в двумерных лазерных алгоритмах SLAM;
Huawei workshop *Сен 2020* Correlation filter for 2D laser indoor scans;
РЖД НИИАС семинар *Окт 2020* Корреляционный фильтр для двумерных лазерных сканов.

Программа для ЭВМ:
Филатов А.Ю., Кринкин К.В., Программа для вычисления взаимного расположения агентов, выполняющих алгоритм решения задачи SLAM. Номер свидетельства RU 2020610524 Дата публикации: 15.01.2020

Основные публикации по теме работы

Список публикаций в изданиях, входящих в перечень **ВАК**:

1. Филатов Ан. Ю, Кринкин К.В., Филатов Ар.Ю., Гулецкий А.Т., Карташов Д.А. Сравнение современных лазерных алгоритмов SLAM // Известия СПбГЭТУ «ЛЭТИ». 2018. № 7. С.66–73
2. Филатов Ан. Ю, Кринкин К.В., Филатов Ар.Ю., Чен Б., Молодан Д. Методы сравнения качества 2D-SLAM-алгоритмов // Известия СПбГЭТУ «ЛЭТИ». 2018. № 7. С.87–95

Список публикаций, индексируемых в базе данных **SCOPUS**:

1. Filatov An., Krinkin K., Filatov Ar. The scan matchers research and comparison: Monte-carlo, olson and hough //2016 19th Conference of Open Innovations Association (FRUCT). IEEE, **2016**. С. 99-105.
2. Filatov An., Krinkin K., Filatov Ar. 2D SLAM quality evaluation methods //Proceedings of the 21st Conference of Open Innovations Association FRUCT. **2017**. С. 120-126.
3. Filatov An., Krinkin K., Filatov Ar. Data distribution services performance evaluation framework //2018 22nd Conference of Open Innovations Association (FRUCT). IEEE, **2018**. С. 94-100.
4. Filatov An., Krinkin K., Filatov Ar. Evaluation of modern laser based indoor slam algorithms //2018 22nd Conference of Open Innovations Association (FRUCT). IEEE, **2018**. С. 101-106.
5. Filatov An., Krinkin K. Multi-Agent SLAM Approaches for Low-Cost Platforms //2019 24th Conference of Open Innovations Association (FRUCT). IEEE, **2019**. С. 89-95.
6. Filatov An., Krinkin K. A Simplistic Approach for Lightweight Multi-Agent SLAM Algorithm. // International Journal of Embedded and Real-Time Communication Systems (IJERTCS). - **2020** - Т. 1. - № 3. - С. 67-83.
7. Filatov An., Krinkin K., Correlation Filter for 2D Laser Indoor Scans. // Robotics and Autonomous Systems. – **2021**. – in press.

.

Соответствие паспорту специальности

05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

1. Модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования.

3. Модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем.

9. Модели, методы, алгоритмы и программная инфраструктура для организации глобально распределенной обработки данных.

Спасибо за внимание!

Запасные слайды

Таблицы по коэффициентам Спирмана и Кендалла

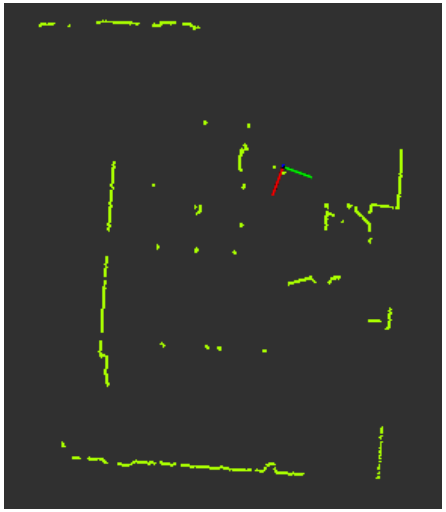
X	Y	rgX	rgY	Q
18.4	5.57	1	3	2
20.6	2.88	2	1	0
21.5	4.12	3	2	0
35.7	7.24	4	4	0
37.1	9.67	5	6	1
39.8	10.48	6	9	3
51.1	8.58	7	5	0
54.4	14.79	8	10	2
64.6	10.22	9	7	0
90.6	10.45	10	8	0

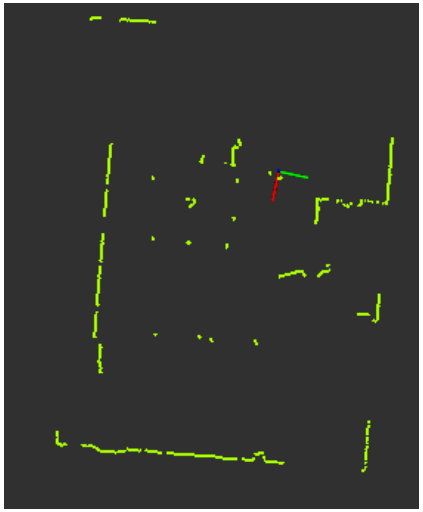
$$k_P = 0.68$$

$$k_S = 0.81$$

$$k_K = 1 - \frac{4 \cdot 8}{10(10-1)} = 0.64$$

Сравнение коэффициентов корреляции для лазерных сканов



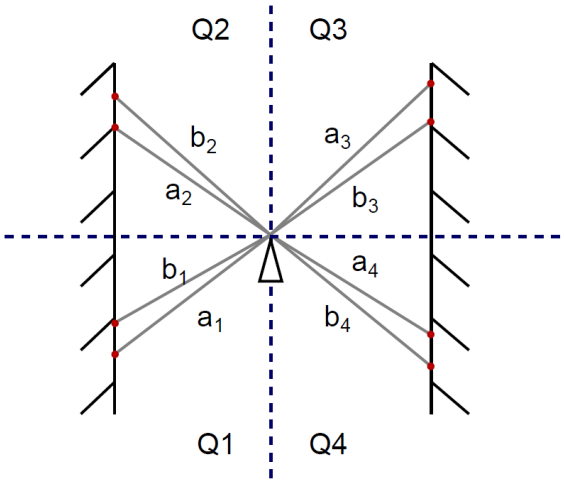


$$k_P = 0.9784$$
$$k_S = 0.9611$$
$$k_K = 0.9498$$

Данные по датасету от MIT	Коэффициент Пирсона	Коэффициент Спирмана	Коэффициент Кендалла
Средний % отброшенных сканов	54.70	51.14	53.89
Среднее время фильтрации, мс	0.12	0.21	0.19

Быстрый алгоритм определения прямых линий

$score \leftarrow 0$
for all $point \in laser\_scan$ **do**
  if $point \in quarter_1$ or $quarter_3$ **then**
    $score \leftarrow score + \text{sign}(point - point\_next)$
  else
    $score \leftarrow score - \text{sign}(point - point\_next)$
  end if
end for
$score \leftarrow score/\text{sizeof}(laser\_scan)$



$$a_1 > b_1 \;\&\; a_2 < b_2 \;\&\; a_3 > b_3 \;\&\; a_4 < b_4$$

Сравнение с Google Cartograher

Выигрыш во времени работы достигается за счёт:

- Отказа от графовой структуры алгоритма
- Применения стохастического метода скан матчинга, который в отличие от метода градиентного спуска не останавливается на локальном экстремуме

24 скана/секунду – это показатель близкий к «реальному времени» (30 сканов/секунду). Поскольку он достигается на низкопроизводительной плате прошлого поколения, это гарантирует, что на платах нового поколения он будет работать быстрее «реального времени»

Реальное время

Алгоритм работает в “реальном времени”, если он, подобно человеческому глазу, обрабатывает от 30 сканов в секунду.

Такая характеристика напрямую зависит от мощности вычислительного устройства, поэтому алгоритм может работать в “реальном времени” на одном устройстве и не успевать это делать на другом

TTX Raspberry Pi

1. Raspberry Pi 3 Model B (Процессор Broadcom BCM2837 **1.1 GHz**, оперативная память LPDDR2 **1GB**, операционная система Ubuntu Xenial x64);
2. Raspberry Pi 3 Model B+ (Процессор Quad Core **1.2GHz** Broadcom BCM2837B0, оперативная память LPDDR2 **1GB**, операционная система Ubuntu Xenial x64);
3. Raspberry Pi 4 Model B (Процессор Quad Core **1.5GHz** Broadcom BCM2711, оперативная память LPDDR4 **2GB**, операционная система Ubuntu Xenial x64);
4. Персональный компьютер (Процессор: Intel Core i7-860 **4x2.8GHz**, оперативная память DDR3 **8GB**, операционная система Ubuntu Xenial x64).

Известные наборы данных

Название	Истинная траектория	Датчики	Окружение
MIT Stata	+	Лидар, камера, одометрия	Помещение
2D Cartographer Backpack – Deutsches Museum	-	Лидар, одометрия	Помещение
PR2 – Willow Garage	-	Лидар, одометрия	Помещение
KITTI	+ (созданная из лидарных данных)	Лидар, камера, одометрия	Улица
COLMAP	-	камера	Улица

Способы определения точности алгоритма SLAM

- СКО построенной траектории от истинной. Чаще всего применяется в симуляциях.
- Качественная оценка карты. Невозможно автоматизировать.
- Количественная оценка карты[5]. Эвристики. Низкая корреляция с количественной характеристикой СКО.

5. Filatov An., Krinkin K., Filatov Ar. 2D SLAM quality evaluation methods //Proceedings of the 21st Conference of Open Innovations Association FRUCT. **2017**. С. 120-126.