

СПИСОК ЛИТЕРАТУРЫ

1. Смоленцев Н. К. Основы теории вейвлетов. Вейвлет-анализ в Matlab. 4-е изд. М.: ДМК Пресс, 2014.
2. Малла С. Вейвлеты в обработке сигналов / пер. с англ. М.: Мир, 2005. 671 с.
3. Чуи К. Введение в вейвлеты / пер. с англ. М.: Мир, 2001. 412 с.
4. Сергиенко А. Б. Цифровая обработка сигналов: учеб. пособие. 3-е изд. СПб.: БХВ-Петербург, 2011. 768 с.
5. Huang N. E. The empirical mode decomposition and the Hilbert spectrum for non-linear and non-stationary time series analysis // Proc. of the Royal Society of London. 1998. Vol. 454. P. 903–995.
6. Мэтьюз Д. Г., Финк К. Д. Численные методы. Использование Matlab. М.: Вильямс, 2001. 720 р.
7. Вержбицкий В. М. Основы численных методов. М.: Высш. шк, 2002. 840 с.
8. Калиткин Н. Н. Численные методы. М.: Наука, 1978. 512 с.

D. M. Klionskiy, V. V. Geppener
Saint Petersburg Electrotechnical University «LETI»

APPLICATION OF TIME-FREQUENCY HILBERT-HUANG DISTRIBUTION FOR ANALYZING OSCILLATING SIGNALS

Is devoted to studying the time-frequency Hilbert-Huang spectrum for adaptive signal processing. Time-frequency distribution is constructed using the components extracted from the original signal using the empirical mode decomposition (EMD). Empirical modes are used for calculating the analytic signal and further the time-frequency spectrum. We compute full phase of the signal, which is further used for calculating the instantaneous frequency. The paper also discusses empirical mode normalization for calculating the instantaneous frequency and the necessary preprocessing. Hilbert-Huang spectrum may be used in order to extract amplitude and frequency modulations in signals, classify signals, and find areas of energy concentration. This spectrum may be applied in order to calculate instantaneous energy density and marginal spectrum, which depends only on frequency and can be considered as an analogue of Fourier spectrum for non-stationary signals.

Time-frequency distribution, Hilbert-Huang spectrum, empirical mode decomposition, instantaneous frequency, empirical mode, oscillating signal

УДК-519.688

Ар. Ю. Филатов, Ан. Ю. Филатов, А. Т. Гулецкий, Д. А. Карташов, К. В. Кринкин
Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Сравнение современных лазерных алгоритмов SLAM

Дается полное описание общей методологии алгоритмов SLAM (одновременное построение карты и определение на ней собственного местоположения), основанных на данных 2D-сканирования реальных помещений. Существующие подходы могут быть проклассифицированы в рамках представленной модели, а также возможно провести их сравнение. Были рассмотрены следующие алгоритмы: Google Cartographer [1], GMapping [2], tinySLAM [3]. Согласно их оценке, Cartographer и GMapping более точны, чем tinySLAM, а Cartographer – самый надежный из алгоритмов. Рассмотрен вопрос количественного и качественного оценивания результатов работы алгоритма одновременного построения карты и определения на ней местоположения. Количественная оценка предполагает наличие истинной траектории движения робота во время записи тестовых данных. Качественная оценка может быть выполнена только человеком: необходимо оценить консистентность карты, наличие на ней самопересечений, разрывов прямых линий и прочее.

2D лазерный SLAM, cartographer, gmapping, tinySLAM, groundtruth

Одновременное ориентирование и построение карты (англ. вариант, SLAM) – это проблема построения карты и в то же время локализации робота внутри этой карты без предварительного знания окружения и положения мобильной плат-

формы. Существуют различные подходы SLAM, основанные на данных от разных типов датчиков. Популярная конфигурация входных данных для мобильных платформ обеспечивает:

– лазерный скан – набор расстояний от платформы до обнаруженных препятствий.

– одометрию – информацию о перемещении платформы.

Первый тип входных данных представляет собой 2D-срез среды, которая содержит в себе меньше информации, чем трехмерное представление. Поэтому обычно требуется меньшее количество вычислений, которое больше подходит для недорогих мобильных платформ, где ограничены вычислительные ресурсы.

Проблема SLAM четко определена, но алгоритм ее решения для произвольной среды еще не был предложен [4]. Предполагается, что некоторые внедренные подходы используются в наружной среде, другие – в помещениях, где преобладают прямые линии.

Несмотря на различия в 2D-алгоритмах SLAM, они используют общие идеи и структуры данных. Информация об окружающей среде хранится как сетка вероятностной занятости – двумерный массив значений, которые представляют собой вероятность соответствующих ячеек в пространстве быть занятыми препятствием. Положение платформы (поза робота) состоит из координат и угла поворота. Комбинация позы робота и соответствующей карты называется состоянием мира. Алгоритмы обычно используют компонент сопоставления сканов, который корректирует позу робота, оцениваемую с помощью необработанной одометрии, сопоставляя данные лазерного сканирования и предыдущие сканы. Этот компонент можно реализовать несколькими способами: упорядочить точки сканирования [5], найти наилучшую корреляцию между полученным сканом и картой [6] и т. д. На рис. 1 показана схема высокого уровня типичного метода вероятностного SLAM, основанного на лазерном сканировании:

– sensor 1 и sensor 2 обеспечивают данные одометрии и лазерного сканирования соответственно;

– tracking объединяет одометрию с предыдущей оценкой положения робота;

– fMap преобразует заданное сканирование в позу робота (обычно такое преобразование выполняется с помощью счетчика сканирования с учетом уже построенной карты);

– Sensor fusion объединяет две оценки позы;

– \oplus добавляет лазерный скан к карте.

На рис. 1 представлен вид алгоритмов на основе цикла предсказания и коррекции, который состоит из следующих этапов: прогнозирование позы (tracking) робота, коррекция данных с помо-

щью экстероцептивных датчиков (fMap + sensor fusion) и обновление карты (\oplus).

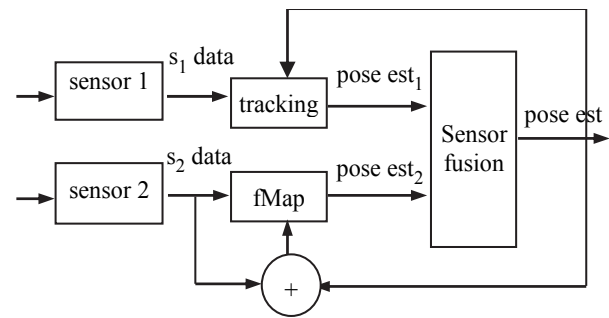


Рис. 1

В популярных методах 2D-SLAM можно было найти следующие идеи:

- однопредположительное представление карты;
- многопредположительное представление карты;
- представление карты в виде графа.

Состояние мира можно рассматривать как случайную переменную с неизвестной плотностью распределения, обновляемую при появлении новых данных от датчиков [7]. Простое одиночное отслеживание гипотез подразумевает оценку и обновление только наиболее вероятного состояния мира. Примерами такого подхода могут служить алгоритмы tinySLAM [3], L-SLAM [8], FastSLAM [9]. Этот метод – не общий, так как среда может содержать шаблоны, которые приводят к плотностям распределения с несколькими максимумами. Если карта хранится в виде сетки, положение скана, уже вставленное в нее, не может быть впоследствии изменено. Слежение за несколькими гипотезами поддерживает несколько состояний (основанных на сетчатых картах), поэтому существует возможность поддерживать такой набор вероятных карт, пока в PDF не станет содержаться единственный максимум. Например, GMapping [2] и DP-SLAM [10] реализуют эту идею с помощью фильтра частиц. Он обновляет каждое состояние мира (частицу) независимо посредством цикла предсказания и коррекции, описанного выше. Цель фильтра частиц – это аппроксимация плотности распределения с помощью набора элементов (состояний мира) из его области. Набор должен периодически обновляться с помощью операции повторной дискретизации, чтобы отразить изменения в плотности распределения. Он заполняет новый набор частиц в окрестности наиболее вероятных состояний. Однако передискретизация страдает от истощения частиц: существует вероятность отклонения от истинного состояния мира. Например, чтобы справиться

с этой проблемой, GMapping пытается увеличить время перед повторной выборкой. Другая стратегия заключается в добавлении новых частиц без учета существующих.

По сути, множественное отслеживание гипотез содержит несколько конфигураций собранных лазерных сканов, что увеличивает разнообразие карт. Еще один метод – сохранение карты в виде графа, который позволяет изменять положения уже интегрированных сканов для достижения консистентности [11].

Граф может быть структурирован следующим образом: вершины содержат комбинацию позиции робота и наблюдения, ребра – преобразования между вершинами. Такое представление позволяет корректировать преобразование (предоставляемое сопоставителем сканов) позже, при оптимизации графа. Карта должна быть топологически достоверной, чтобы правильно уточнить эту оптимизацию. Топологическая корректность может быть нарушена, если робот не может обнаружить себя в уже посещенной вершине (проблема замыкания цикла). Оптимизация в конечном итоге выполняется, когда новый цикл закрывается на графике. Цель этого процесса – уменьшить и распределить накопленную ошибку среди ребер. Для оптимизации можно использовать, например, библиотеку g2o [12]. Примерами методов, использующих графическую карту, служат Google Cartographer [1] и Graph-Based SLAM [13].

Обзор методов SLAM на основе 2D-лазерного сканирования представлен в [14]. Было оценено пять алгоритмов: примерами отслеживания одиночных гипотез служили HectorSLAM и tinySLAM, многогипотезные SLAM были представлены GMapping, а графовые SLAM – KartoSLAM и LagoSLAM. Точность методов SLAM оценивалась нормированным расстоянием между полученными занятыми ячейками на карте и истинными картами. Первоначально tinySLAM, как утверждается, имеет наибольшую ошибку по сравнению с другими, в то время как GMapping показывает лучшую точность. Эти результаты получены на моделированных (12.2×11.7 м) и небольшого реального пространства в помещении (4.6×4 м) областях.

Цель этой статьи – сравнить три алгоритма SLAM: улучшенный tinySLAM [15], оригинальный GMapping и новый Google Cartographer. Они были выбраны в качестве воплощений описанных идей. Кроме того, пределы применения Google

Cartographer оцениваются с использованием набора данных, не упомянутого в представленном документе.

Описание алгоритмов. 1. *tinySLAM*. Как уже упоминалось, tinySLAM отслеживает одну гипотезу о реальном мире. Метод основан на цикле предсказания и коррекции, поэтому он имеет компактную и простую реализацию.

Карта хранится как сетка ячеек. Каждая ячейка сохраняет вероятность быть занятой для части среды, которой она соответствует. Вероятность обновляется по следующей формуле:

$$\text{map}_{n+1}(x, y) = (1 - q) \text{map}_n(x, y) + q \cdot \text{value},$$

где $\text{map}_n(x, y)$ – оценка занятости карты на предыдущем шаге; value – просмотренное значение ячейки; 1 – если лазер нашел препятствие в этой ячейке, 0 – иначе; q – доверие к измерениям.

Поза уточняется с помощью сопоставителя сканов на основе метода Монте-Карло. Он пытается добавить нормально распределенные (нулевые средние, настраиваемые дисперсии) значения для каждой координаты позы, чтобы найти новое положение, которое ведет к лучшему соответствию между картой и последним полученным сканом. Соответствие оценивается как сумма вероятностей занятости ячеек, содержащих точки лазерного скана.

Несколько эвристик реализованы tinySLAM, так как такой компонент сопоставления сканов недостаточно устойчив. Размытие стен карты увеличивает вероятность появления частично занятых ячеек в окрестности точки скана. Исходная реализация также интерполирует данные лазерного скана для увеличения количества точек в лазерном скане и применяет их для сопоставления.

Другая эвристика предлагается в [15]. Она добавляет лазерные сканы к карте с более низким «качественным» значением, если поза робота была изменена сопоставителем сканов. Измененная модель ячейки, введенная в статью, возвращает среднее значение вероятности заполнения ячейки.

2. *Gmapping*. Картирование основано на фильтре частиц Рао–Блэквелла (РБФЧ) [16], который оценивает апостериорное значение вероятности $p(x_{1:t} | z_{1:t}, u_{0:t})$ представляющее потенциальную траекторию робота $x_{1:t}$, используя наблюдения $z_{1:t}$ и данные одометрии $u_{0:t}$. Апостериорное значение вероятности аппроксимируется множеством точек (частиц) с соответ-

ствующими вероятностями (весами). Частица с максимальным весом рассматривается как реальное состояние мира. Вес частицы обновляется мерой соответствия между новым сканером и картой, оцененной с помощью сопоставителя сканов.

GMapping использует теорию Рао–Блэквелла, при этом:

- уменьшается неопределенность по положению робота на этапе прогнозирования цикла предсказания и коррекции;

- проблема истощения частиц решается выборочной передискретизацией.

Модель ячейки сетки, используемая GMapping, сохраняет средние значения занятости и положения всех препятствий, обнаруженных лазерным дальномером в этой ячейке.

GMapping использует своего рода метод градиентного спуска для соответствия сканированиям. На каждой итерации проверяется несколько предопределенных направлений. В качестве начальной позиции для следующей итерации выбирается позиция вдоль направления с максимальной совпадающей оценкой. Оценка сканирования рассчитывается следующим образом:

$$\text{score}(\text{scan}, \text{map}) = \sum_{p \in \text{scan}} e^{-\frac{1}{\sigma} d(p, \text{map})^2},$$

где p – точка скана; $d(p, \text{map})$ – минимальное расстояние между p и препятствием на карте; σ – предустановленное значение дисперсии.

Обычно gMapping требует менее 80 частиц даже в достаточно больших окружениях (250×250 м) для построения точных карт (по умолчанию используется 30) [2].

3. *Google Cartographer*. Программа (система) хранит карту окружения в виде графа, где каждая вершина представляет миникарту, и скана, полученного после создания соответствующей миникарты. Ребра представляют собой преобразования между соответствующими миникартами. Таким образом, дополнительно появляется шаг оптимизации, сохраняющий карту консистентной.

Каждая ячейка миникарты характеризуется вероятностью быть занятой. Она обновляется в случае, если миникарта еще не была полностью сконструирована. Следующая формула представляет правило обновления значения вероятности ячейки:

$$M_{\text{new}}(\text{cell}) = \text{odds}^{-1} \left\{ \text{odds} \left[M_{\text{old}}(\text{cell}) \text{odds}(p_{\text{hit}}) \right] \right\},$$

где $M_{\text{old}}(\cdot)$ – предыдущее значение занятости ячейки; p_{hit} – вероятность ячейки быть занятой;

$$\text{odds}(p) = \frac{p}{1-p}.$$

Механизм сопоставления сканов в свою очередь заключается в минимизации функционала стоимости. Таким образом, процесс сопоставления сканов проходит через минимизацию следующего функционала:

$$\arg \min_{\xi} \sum_{k=1}^K \left[1 - M_{\text{sm}}(T_{\xi} h_k) \right]^2,$$

где $M_{\text{sm}}(x)$ – значение ячейки x , сглаженной значениями окрестности; h_k – ячейка, включающая точку лазерного скана; T_{ξ} – матрица трансформации, сдвигающая точку h_k на вектор смещения $\xi = (\xi_x, \xi_y, \xi_{\theta})^T$.

Такая минимизация осуществляется с помощью алгоритма полного перебора, а именно с применением метода ветвей и границ. Для быстрой работы процесса используется библиотека Ceres [17].

Проблема оптимизации также решена с помощью алгоритма полного перебора. Требуется искать не только одно значение ξ , но и множество всех подходящих преобразований между вершинами. Библиотека Ceres также используется для их поиска. Оптимизации занимают много времени, поэтому эта работа выполняется каждые несколько секунд параллельно основному процессу и исправляет уже зарегистрированные результаты. Исходя из вышеуказанного, алгоритм Cartographer нельзя назвать строго алгоритмом реального времени.

Сравнение. Точность алгоритма SLAM может быть оценена сравнением выходной траектории с истинной. Для верной оценки применимости метода SLAM сравнение должно проводиться на наборах данных, собранных в реальных условиях.

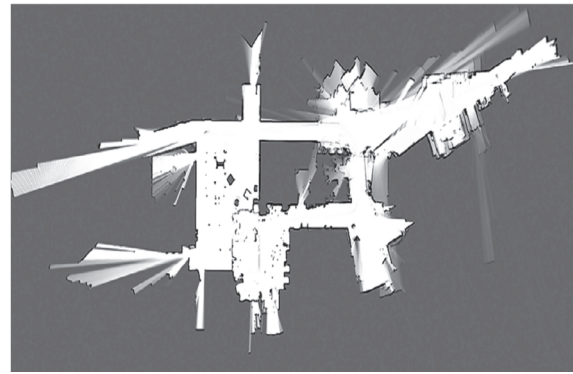
Набор данных MIT [18] относится к полученным с помощью лазерного дальмера и предоставляющим истинные траектории движения. Эти данные можно извлечь методом локализации на

плане этажа с набором данных. Он содержит 84 последовательности, но некоторые из них – многоэтажные (в отличие от Cartographer, как tinySLAM, так и GMapping не могут работать в таких средах), а другие не снабжаются соответствующими планами этажей. Таким образом, для тестирования рассмотренных алгоритмов SLAM было выбрано 11 последовательностей.

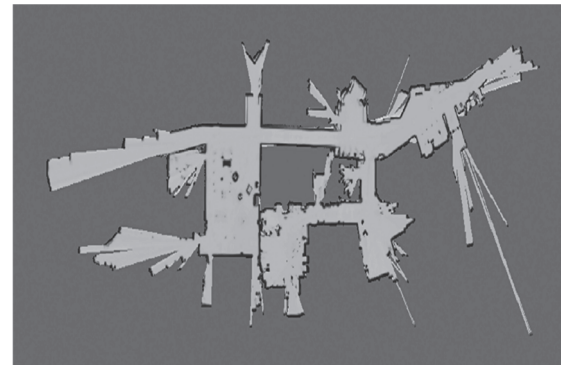
Набор данных Willow Garage [19] не представляет истинных траекторий движения, поэтому точность алгоритмов не может быть оценена количественно. В этом случае алгоритмы SLAM можно качественно сравнить, визуально сопоставляя выходные карты. Авторы Cartographer используют этот набор данных для демонстрации эффективности алгоритма. Для тестирования tinySLAM и Gmapping была выбрана одна тестовая последовательность.

Оба рассмотренных датасета собраны с использованием робота PR2 с лазерным дальномером Hokuyo UTM-30LX.

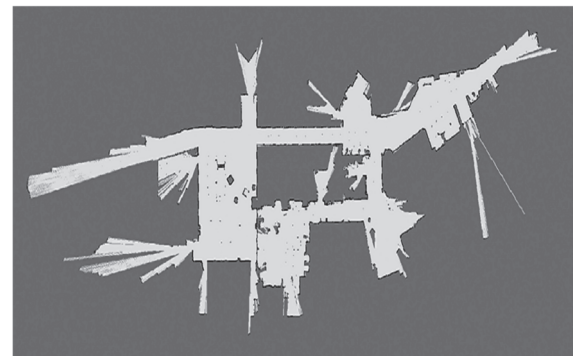
Поскольку все алгоритмы – вероятностные, их выход может варьироваться от одного запуска к другому. Для оценки точности алгоритмов SLAM вычисляется среднеквадратичная ошибка (СКО) между выходными и истинными траекториями. Эта оценка была выполнена в наборе данных MIT. Траектории сравнивались при помощи инструмента оценки SLAM TUM. Все алгоритмы были протестированы «как есть», т. е. ни один из параметров ни одного алгоритма не был изменен для определенной последовательности.



а



б



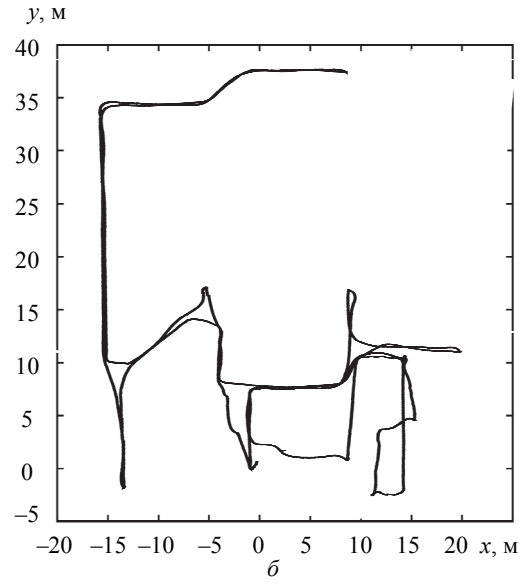
в

Рис. 2

Последовательность	Длина, м	СКО траекторий, м			
		Gmapping	tinySLAM	Cartographer (онлайн)	Cartographer (офлайн)
2011-01-19-07-49-38	68	0.216 ± 0.012	1.280 ± 0.640	0.188 ± 0.023	0.191 ± 0.001
2011-01-20-07-18-45	76	0.219 ± 0.012	0.245 ± 0.045	0.219 ± 0.002	0.221 ± 0.001
2011-01-21-09-01-36	87	0.212 ± 0.029	0.242 ± 0.005	0.217 ± 0.003	0.205 ± 0.001
2011-01-24-06-18-27	87	0.290 ± 0.035	0.245 ± 0.006	0.217 ± 0.001	0.217 ± 0.001
2011-01-25-06-29-26	109	0.208 ± 0.008	0.260 ± 0.005	0.232 ± 0.001	0.232 ± 0.002
2011-01-27-07-49-54	94	0.266 ± 0.012	0.620 ± 0.030	0.266 ± 0.004	–
2011-01-28-06-37-23	145	2.388 ± 1.949	2.280 ± 0.750	0.360 ± 0.069	0.354 ± 0.004
2011-03-11-06-48-23	245	0.365 ± 0.208	0.860 ± 0.390	1.152 ± 0.601	1.348 ± 0.001
2011-03-18-06-22-35	80	0.145 ± 0.023	0.103 ± 0.008	0.145 ± 0.021	–
2011-04-06-07-04-17	95	0.190 ± 0.002	0.343 ± 0.025	0.201 ± 0.002	–
2011-10-20-11-38-39	264	0.352 ± 0.003	5.486 ± 2.603	2.217 ± 0.021	–



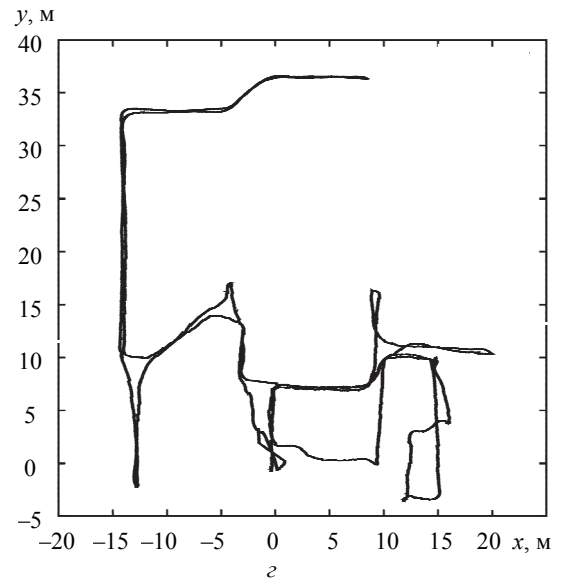
a



б



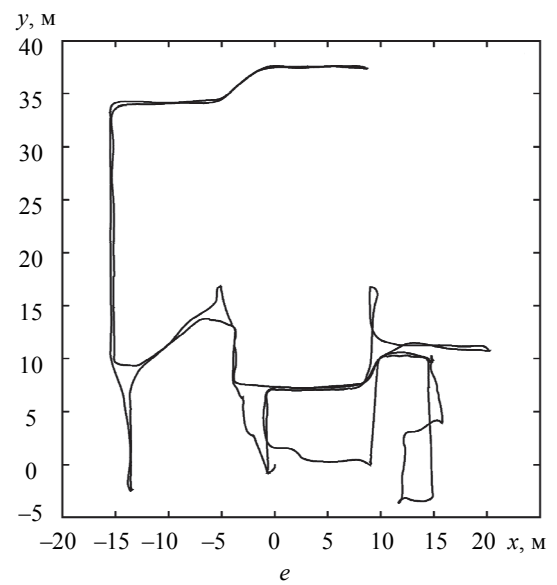
в



г



д



е

Рис. 3

Средние и дисперсионные значения СКО в течение нескольких прогонов представлены в таблице. Для Cartografer в ней содержатся два столбца, поскольку он может выполнять окончательную оптимизацию траектории робота после обработки всей последовательности данных. Столбец «Cartografer (офлайн)» содержит значения RMSE после окончательной оптимизации, а в столбце «Cartografer (онлайн)» отображается RMSE данных траектории, собранных в процессе выполнения. Однако Cartografer не смог оптимизировать траектории для некоторых последовательностей данных (знак «←» в таблице), что может быть вызвано ошибками в проверенной реализации алгоритма.

Результаты показывают, что, вообще говоря, GMapping и Cartographer имеют сравнимую точность на коротких траекториях (менее 100 м). В то же время Cartographer имеет более низкую дисперсию результатов, т. е. его результат более предсказуем. Однако Cartographer может ошибаться на последовательностях с длинными прямыми залогами, например в последовательности «2011-03-11-06-48-23» (рис. 2, а). Множественное отслеживание гипотез, реализованное в GMapping, обычно дает гораздо более точные результаты на входе (рис. 2, б). Метод tinySLAM также не смог построить согласованную карту на большинстве таких последовательностей, но иногда он дает лучшее качество вывода, чем Cartographer (рис. 2, в). Успешное завершение окончательной оптимизации траектории в алгоритме Cartographer оказывает небольшое влияние на значение RMSE, но значительно уменьшает дисперсию, которая может быть полезна, если качество карты окружения более важно, чем точность траектории.

Качество карт, построенных по рассматриваемым алгоритмам, оценивалось в последовательности «2011-08-04-23-46-28» из набора данных Willow Garage. Карты и траектории, представленные на рис. 3, – лучшие из полученных после нескольких прогонов алгоритмов. Согласно рис. 3 (а – карта

Gmapping, б – траектория Gmapping; в – карта Cartographer, г – траектория Cartographer; д – карта tinySLAM, е – траектория tinySLAM) все проверенные алгоритмы способны построить достаточно согласованную карту, но карта, построенная tinySLAM (рис. 3, д), имеет больше артефактов, чем другие. Стоит отметить, что для tinySLAM потребовалось наибольшее количество прогонов для получения согласованной карты, что означает ее меньшую надежность. Тем не менее, все траектории вывода достаточно близки друг к другу.

Сравнивались три реализации популярных алгоритмов SLAM, использующие лазерный дальномер: tinySLAM, GMapping и Cartographer. Оценка проводилась на двух наборах данных, собранных в реальных условиях в помещении. Алгоритмы выполнялись с использованием конфигураций по умолчанию, хотя изменение некоторых параметров могло повысить точность, но это противоречит идее универсального (произвольно применимого) метода SLAM.

Было показано, что tinySLAM обеспечивает наибольшую среднеквадратическую ошибку на длинных последовательностях, поскольку она отслеживает только одну гипотезу и накапливает ошибки во время оценки. Из-за своей стохастической природы tinySLAM может неправильно оценивать состояние мира, что приводит к ошибкам на карте, которая не может быть исправлена. Как и ожидалось, GMapping обеспечивает меньшую ошибку, чем tinySLAM, поскольку он одновременно поддерживает множественную гипотезу о мировом состоянии и использует нестационарный сканер.

Cartographer и GMapping имеют сопоставимые значения RMSE для большинства входных последовательностей. Cartographer обычно имеет более низкое значение дисперсии ошибки, поэтому он более устойчив. Однако он менее точен при других последовательностях, поэтому проверенная реализация не универсальна.

СПИСОК ЛИТЕРАТУРЫ

1. Hess W., Kohler D., Rapp H. Real-time loop closure in 2D LIDAR SLAM // Proc. of IEEE Intern. Conf. on Robotics and Automation (ICRA) Stockholm: IEEE Robotics, 2016. P. 1271–1278.
2. Grisetti G., Stachniss C., Burgard W. Improved techniques for grid mapping with Rao-Blackwellized

particle filters // IEEE Trans. on Robotics. 2007. № 1. P. 34–46.

3. Steux B., Hamzaoui O. E. TinySLAM: A SLAM algorithm in less than 200 lines C-language program // Proc. of the 11th Intern. Conf. on Control Automation Robotics Vision. Washington: IEEE Robotics, 2010. P. 1975–1979.

4. Siegwart R., Nourbakhsh I. R., Scaramuzza D. Introduction to Autonomous Mobile Robots. Cambridge: The MIT Press, 2011.
5. Lu F., Milios E. Robot pose estimation in unknown environments by matching 2D range scans // Proc. of IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition. Milan: IEEE Robotics, 1994. P. 935–938.
6. Konolige K., Chou K. Markov localization using correlation // Proc. of the 17th Intern. Joint Conf. on Artificial Intelligence (IJCAI). Seattle, Morgan: Kaufmann Publishers Inc., 1999. P. 1154–1159.
7. Thrun S., Burgard W., Fox D. Probabilistic robotics (Intelligent robotics and autonomous agents). Cambridge: The MIT Press, 2005.
8. Zikos N., Petridis V. L-SLAM: Reduced dimensionality FastSLAM with unknown data association // Proc. of IEEE International Conf. on Robotics and Automation. Shanghai: IEEE Robotics, 2011. P. 4074–4079.
9. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges / M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit // Proc. of the 18th intern. joint conf. on Artificial intelligence. Mexico: IEEE Robotics, 2003. P. 1151–1156,.
10. Eliazar A. I., Parr R. DP-SLAM 2.0 // Proc. of the IEEE Intern. Conf. on Robotics and Automation. New Orleans: IEEE Robotics, 2004. P. 1314–1320.
11. Lu F., Milios E. Globally consistent range scan alignment for environment. Mapping: Kluwer Academic Publishers, 1997.
12. G2O: A general framework for graph optimization / R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard // Proc. of IEEE Intern. Conf. on Robotics and Automation. Boston: IEEE Robotics, 2011. P. 3607–3613.
13. A Tutorial on Graph-Based SLAM / G. Grisetti, R. Kummerle, C. Stachniss, W. Burgard // IEEE Intelligent Transp. Syst. Mag. 2011. Vol. 2. P. 31–43.
14. Santos J. M., Portugal D., Rocha R. P. An evaluation of 2D SLAM techniques available in Robot Operating System // Proc. of the IEEE Intern. Symposium on Safety, Security, and Rescue Robotics (SSRR). Linkoping: IEEE Robotics, 2013. P. 1–6.
15. Huletski A., Kartashov D., Krinkin K. TinySLAM improvements for indoor navigation // IEEE Intern. Conf. on Multisensor Fusion and Integration for Intelligent Systems, 2016. Daegu: IEEE Robotics, 2016. P. 144–152.
16. Rao-Blackwellised particle filtering for dynamic Bayesian networks / A. Doucet, N. Freitas, K. P. Murphy, S. J. Russell // Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers Inc, 2000. P. 176–183.
17. Agarwal S., Mierle K. and others. Ceres solver. URL: <http://ceres-solver.org> (дата обращения 21.03.18).
18. The MIT Stata Center dataset / M. Fallon, H. Johannsson, M. Kaess, J. Leonard // The Intern. J. of Robotics Research. Vol. 32, № 14. 2013. P. 1695–1699.
19. Mason J., Marthi B. An object-based semantic world model for longterm change detection and semantic querying // Proc of the IEEE/RSJ Intern. Conf. on Intelligent Robots and Syst. (IROS). Vilamoura: IEEE Robotics. 2012. P. 3851–3858.

Ar. Yu. Filatov, An. Yu. Filatov, A. T. Huletski, D. A. Kartashov, K. V. Krinkin
Saint Petersburg Electrotechnical University «LETI»

COMPARISON OF MODERN SLAM LASER ALGORITHMS

Gives a complete description of the general methodology of SLAM algorithms (simultaneous construction of maps and definition of its own locations on it), based on 2D-scanning of real rooms. Existing approaches can be classified within the framework of the presented model, and it is also possible to compare them. The following algorithms were considered: Google Cartographer, GMapping, tinySLAM. According to their assessment, the Cartographer and GMapping are more accurate than tinySLAM, and the Cartographer is the most reliable of the algorithms. The problem of quantitative and qualitative estimation of the results of the algorithm of simultaneous construction of maps and determination on it is considered. The quantitative estimation assumes presence of a true trajectory of movement of the robot during record of the test data. A qualitative assessment can be performed only by a person: it is necessary to assess the consistency of the map, the presence of self-intersections, discontinuities of straight lines, and so on.

2D laser SLAM, Cartographer, gmapping, tinySLAM, groundtruth