

## Алгоритм построения трехмерной карты окружающей среды с использованием камеры глубины

А. Ортиз-Гонзалес\*, В.И. Кобер\*\*, В.Н. Карнаухов\*\*, М.Г. Мозеров\*\*

\*Департамент компьютерных наук, Центр научных исследований и высшего образования, Энсенада, 22860, Мексика

\*\*Институт проблем передачи информации, Российская академия наук, Москва, 127051, Россия

Поступила в редколлегию 21.11.2019

**Аннотация**—В данной работе предложен алгоритм трехмерной реконструкции окружающей среды, который способен создавать точные трехмерные карты в режиме реального времени с помощью камеры глубины RGB-D. Алгоритм может быть использован в автономной мобильной робототехнике, где роботу необходимо локализовать себя в неизвестных средах путем обработки бортовых датчиков без внешних систем ссылок, таких как система глобального позиционирования. В работе проведен анализ эффективности применения для решения задачи различных комбинаций современных детекторов и дескрипторов визуальных признаков. Для сопоставления облаков точек между последовательными кадрами используется метод итеративных ближайших точек. Для улучшения качества трехмерной реконструкции предложена адаптивная схема вычисления оценки положения камеры. Результаты моделирования на известных базах данных показали, что предлагаемый алгоритм способен эффективно обрабатывать сложные сцены в реальном масштабе времени.

**КЛЮЧЕВЫЕ СЛОВА:** трехмерная реконструкция, метод итеративных ближайших точек, камера глубины, одновременная локализация и построение карты.

### ВВЕДЕНИЕ

В области робототехники и машинного зрения разработано много методов построения плотных трехмерных карт на основе камер глубины, которые являются частью задачи одновременной локализации и построения карты (англ. Simultaneous localization and mapping SLAM) [1]. Локализация камеры и построение карты являются взаимосвязанными процессами, так как для точной локализации камеры в окружающей среде необходимо знать ее карту, а для точного построения карты необходимо знать положение камеры на ней. Методы визуальной системы SLAM приобрели большую популярность в последнее десятилетие благодаря возможности системы получать метрическую, яркостную, цветовую, текстурную информацию об окружающей среде и создавать точные и реалистичные трехмерные карты [2]. Однако алгоритмы построения карты окружающей среды могут быть неточными из-за недостаточного разрешения камеры, неравномерного освещения, наличия поверхностей с недостаточной текстурой, размытых изображений из-за быстрых движений камеры.

Для решения задачи SLAM наиболее популярными являются два подхода [3]: на основе расширенной фильтрации Калмана (EKF) и на основе пакетной оптимизации графов. Отметим, что алгоритмы SLAM на основе графов в настоящее время являются самыми эффективными с точки зрения точности и скорости [4]. В целом, система SLAM на основе графов, использующая камеры RGB-D, может быть разбита на два модуля [5]: Frontend и Backend. Frontend обрабатывает данные RGB-D для вычисления движения робота. Сначала выделяются ключевые точки для текущего кадра RGB-D. Затем эти точки сопоставляются с ключевыми точками

предыдущего кадра, и ложные соответствия (выбросы) обнаруживаются и отбрасываются с использованием алгоритма RANSAC [6]. Относительное движение вычисляется с использованием набора всех прошедших проверку соответствий и алгоритмов трехмерной регистрации, таких как итеративные алгоритмы ближайших точек (ICP) [7], [8], [9]. ICP алгоритм позволяет сопоставлять трехмерные геометрические модели, сводя к минимуму расстояние между точками из двух кадров на основе метода наименьших квадратов, выходными данными алгоритма является матрица поворота и вектор переноса. Для алгоритма характерны две основные проблемы: во-первых, алгоритм не учитывает локальную форму поверхности вокруг каждой точки, а во-вторых, поиск ближайших точек обладает большой вычислительной сложностью. Модуль `Backend` строит граф, который представляет собой геометрические соотношения между позициями робота и ориентирами. Этот граф может быть оптимизирован с использованием дополнительных ограничений, что особенно полезно при повторном посещении роботом уже пройденных участков.

Большинство методов построения трехмерных карт содержит следующие этапы: пространственное выравнивание последовательности кадров данных (регистрация изображений или облаков точек), решение проблемы замыкания цикла и глобальное выравнивание всех кадров данных. На сегодняшний день разработано много точных и эффективных методов, подходов и алгоритмов для решения задачи одновременной навигации и составления карты в статической среде, однако реальные приложения требуют поиска надежного и точного решения данной задачи в динамической среде. Поэтому в данной работе предлагается адаптивный алгоритм SLAM, который позволит построить трехмерную карту окружающей среды в режиме реального времени с помощью недорогой камеры глубины RGB-D.

Статья организована следующим образом: в разделе 1 дан анализ алгоритмов обнаружения и описания визуальных признаков, в разделе 2 описана адаптивная система SLAM, в разделе 3 представлены результаты экспериментов с системами SLAM, и раздел Заключение суммирует наши выводы.

## 1. АНАЛИЗ АЛГОРИТМОВ ВЫДЕЛЕНИЯ И ОПИСАНИЯ ПРИЗНАКОВ

В визуальной системе SLAM одной из основных проблем является выбор ключевых точек на изображении, которые могут быть использованы в качестве надежных ориентиров. Эти точки должны быть инвариантны к повороту, перемещению, масштабированию, проекционным искажениям и неравномерному освещению. На практике, данные требования не всегда выполняются, и возникает проблема поиска соответствия одних и тех же ключевых точек на разных изображениях [10], [11], [12], [13]. Ключевые точки описываются с помощью локальных признаков [14], [15]. Процесс построения признаков состоит из двух этапов: обнаружение точек (детекторы) и описание признаков (дескрипторы). Было предложено много алгоритмов детекторов и дескрипторов, однако, однозначного ответа на вопрос: “Какая комбинация детектор-дескриптор является наиболее подходящей для визуальной системы SLAM?” не существует. В данной работе были протестированы следующие алгоритмы: Shi and Tomasi [16], FAST [17], ORB [18], BRISK [19], STAR [20], BRIEF [21], FREAK [22], LATCH [23], SIFT [24], SURF [25]. Краткое описание этих алгоритмов можно найти в работе [26]. Хотя существует множество баз данных для оценки работы детекторов и дескрипторов, нами была выбрана база данных [27], поскольку она включает в себя общие деформации изображений для тестирования системы SLAM.

Обработка в реальном времени является важным требованием для систем SLAM, поэтому скорость работы алгоритмов детекторов и дескрипторов является важным критерием для их сравнения. Так как временные затраты каждой комбинации детектор-дескриптор зависят от количества признаков, обнаруженных детектором, то вычисляется среднее время на вы-

деление 1000 признаков [28]. Результаты тестирования на выделение одного признака в мкс показаны в Таблице 1. Так как большинство камер RGB-D (например, Microsoft Kinect) работают с частотой 30 Гц, то для работы в режиме реального времени нам нужна комбинация детектора-дескриптора, способная извлекать признаки за время меньшее, чем 33 мс. Все комбинации с временем работы более 20 мкс на извлечения одного признака будут отброшены, они отображаются в Таблице 1 наклонным шрифтом. Эксперименты проводились на стандартном ПК с процессором Intel Core i5-4200U с 8 ГБ оперативной памяти.

**Таблица 1.** Среднее время на выделение одного признака (мкс). Комбинации со средним временем большим, чем 20 мкс помечены наклонным шрифтом.

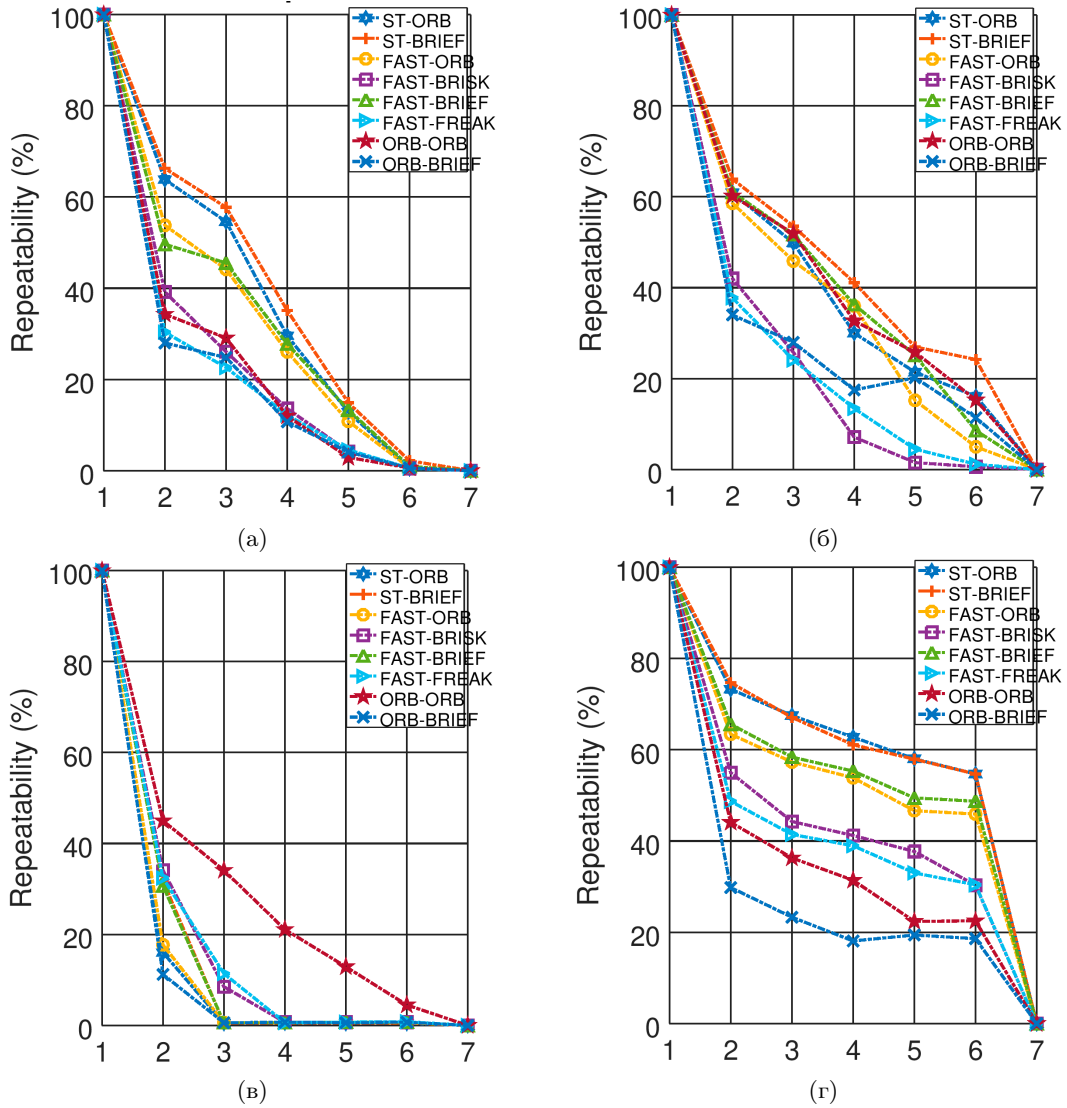
Детекторы	Дескрипторы						
	ORB	BRISK	BRIEF	FREAK	LATCH	SURF	SIFT
Shi and Tomasi	<b>16</b>	<i>23</i>	<b>15</b>	<i>21</i>	<i>53</i>	<i>23</i>	<i>29</i>
FAST	<b>6</b>	<b>16</b>	<b>5</b>	<b>10</b>	<i>46</i>	<i>21</i>	<i>60</i>
ORB	<b>14</b>	<i>21</i>	<b>11</b>	<i>28</i>	<i>50</i>	<i>199</i>	<i>309</i>
BRISK	<i>40</i>	<i>43</i>	<i>38</i>	<i>52</i>	<i>75</i>	<i>93</i>	<i>146</i>
SIFT	$\infty$	<i>144</i>	<i>130</i>	<i>136</i>	<i>162</i>	<i>131</i>	<i>220</i>
SURF	<i>57</i>	<i>70</i>	<i>54</i>	<i>85</i>	<i>92</i>	<i>139</i>	<i>417</i>
STAR	<i>102</i>	<i>95</i>	<i>79</i>	<i>87</i>	<i>124</i>	<i>107</i>	<i>424</i>

Важным требованием для визуальной системы SLAM является способность обнаруживать такие ключевые точки (ориентиры), которые будут стабильными на последовательности кадров. Это свойство характеризуется критерием повторяемости. Повторяемость для каждой комбинации вычисляется подсчетом найденных правильных соответствий на последовательности кадров. Так как после теста скорости у нас остались только бинарные дескрипторы, то будем использовать метрику Хэмминга. Однако расстояние не является окончательным критерием правильного сопоставления, так как дескрипторы могут сильно отличаться из-за высокой размерности пространства признаков. Предложено использовать отношению между ближайшим соседом и вторым ближайшим соседом в пространстве признаков [24]. Для удаления неверных соответствий будем использовать матрицу гомографии, которая дается вместе с базой данных, чтобы перепроектировать найденные совпадения в обратном порядке. Если евклидово расстояние между перепроектированной парой меньше порога, то совпадение считается правильным, в противном случае — неверным. Таким образом, критерий повторяемости можно определить как

$$\text{Repeatability} = \frac{|S_r|}{|S_b|} * 100, \quad (1)$$

где  $S_r$  — множество правильных соответствий при прямом поиске,  $S_b$  — множество соответствий при обратном поиске. Идеальный детектор-дескриптор обнаруживает одни и те же ключевые точки при прямом и обратном поисках, то есть повторяемость равна 100%. Результаты тестирования алгоритмов детектор-дескриптор при проекционных искажениях, смазе, повороте и масштабировании, неравномерном освещении приведены на Рис. 1. По оси абсцисс приведена степень усиления данного искажения.

Из Рис. 1 можно сделать следующие выводы. Из-за низкой повторяемости не будем рассматривать следующие комбинации: FAST-FREAK, FAST-BRISK, ORB-BRIEF. Отметим, что детектор Shi и Tomasi в сочетании с дескрипторами ORB и BRIEF превосходят остальные алгоритмы в большинстве случаев, за исключением поворота и масштабирования, где ORB-ORB является наилучшим алгоритмом. FAST детектор в комбинации с дескрипторами ORB и BRIEF имеет аналогичное поведение с детектором Shi и Tomasi и его комбинациями. Точность



**Рис. 1.** Результаты тестирования алгоритмов детектор-дескриптор при: (а) проекционных искажениях, (б) смазе, (в) повороте и масштабировании, (г) неравномерном освещении.

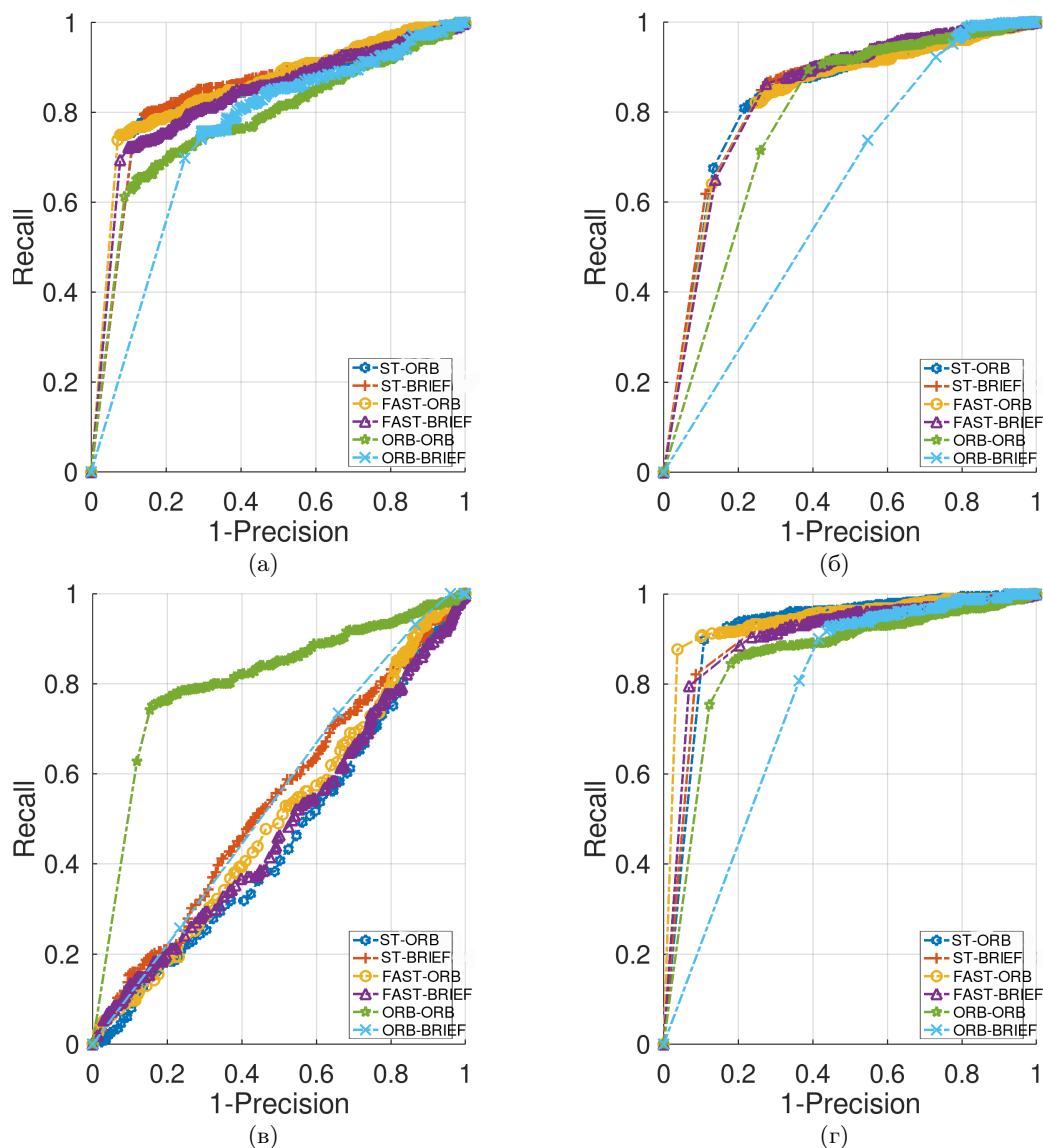
(Precision) и полнота (Recall) обычно используются для характеристики качества процесса сопоставления. Мера точности характеризует относительное количество полученных правильных сопоставлений. Чем больше точность, тем меньше ложных сопоставлений. Мера точности, однако, не дает представление о том, все ли правильные сопоставления получены. Для этого существует так называемая мера полноты. Эти две меры определяются следующим образом [11]

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (3)$$

где TP и FN — истинно-положительные и ложно-отрицательные сопоставления, соответственно. Изменяя порог между правильными и ложными ответами, можно получить различные

значения полноты и точности. Взаимосвязь точности и полноты для различных комбинаций детектор-дескриптор при проекционных искажениях, смазе, повороте и масштабировании, неравномерном освещении приведена на Рис. 2. Рабочая область находится между нулем и единицей, где единица — идеальная работа пары детектор-дескриптор. Такая кривая может быть использована в качестве показателя эффективности комбинация детектор-дескриптор на паре изображений.



**Рис. 2.** Результаты тестирования точности и полноты алгоритмов при: (а) проекционных искажениях, (б) смазе, (в) повороте и масштабировании, (г) неравномерном освещении.

Отметим, что комбинация ORB-BRIEF имеет самую низкую эффективность в большинстве случаев, поэтому в дальнейшем не будем ее использовать. Остальные варианты комбинаций имеют схожую эффективность.

## 2. АДАПТИВНАЯ СИСТЕМА SLAM

Обычная система SLAM состоит из двух модулей: Frontend и Backend [5], [28], [29]. Frontend обрабатывает кадры визуальной информации с целью извлечения признаков и оценки приращения движения робота. В предлагаемой системе этот модуль включает в себя следующие этапы:

1. Предварительная обработка изображений. Изображение, полученное от RGB-D камеры, преобразуется в черно-белое изображение, сглаживается с помощью гауссова ядра. Затем строится гауссова пирамида, чтобы выделять признаки в разных масштабах.
2. Адаптивное выделение признаков. Признаки выделяются на каждом уровне гауссовой пирамиды, используя FAST детектор. Порог для детектора адаптивно подбирается таким образом, чтобы обеспечить достаточное количество признаков для каждого кадра. Данные о глубине, соответствующие выделенным признакам, также извлекаются из изображения глубины кадра. Визуальные признаки, не имеющие данных о глубине, отбрасываются. Для того, чтобы обеспечить равномерное описание кадра, каждый уровень пирамиды размечен сеткой  $80 \times 80$  пикселей. В каждой ячейке сетки должно быть не меньше 25 признаков. Для описания признаков используется BRIEF дескриптор.
3. Начальная оценка движения. Чтобы оценить начальное положение кадра в каждый момент времени, будем использовать модель движения с постоянной скоростью.
4. Сопоставление признаков. Ключевые точки, наблюдаемые в предыдущем кадре, проецируются на текущий кадр с использованием начальной оценки положения кадра. Область поиска соответствия ограничивается кругом проекции заданным радиусом. Признаки сопоставляются путем сравнения их дескрипторов с использованием метрики Хэмминга. Производится полный перебор для поиска пары с минимальным значением метрики.
5. Адаптивная оценка движения. Для устранения неверных совпадений признаков, предлагается алгоритм адаптивной оценки движения (см. Рис. 3) с использованием обобщённого итерационного метода ближайших точек (GICP) [9]. На входе алгоритма — два кадра, а на выходе — лучшее относительное преобразование движения между ними.  $\tau_1, \tau_2, \mu$  — параметры алгоритма. В алгоритме есть три возможных пути. Если результат RANSAC [6] является удовлетворительным, тогда принимается полученная относительная оценка движения. Если ошибка большая, то используется GICP алгоритм для уточнения оценки. В третьем случае движение оценивается исключительно с использованием GICP алгоритма на случайном подмножестве совпадений признаков [30].
6. Выбор ключевого кадра. Чтобы граф был относительно разреженным, используются ключевые кадры. Новые ключевые кадры добавляются, когда предыдущий ключевой кадр повернут относительно текущего на угол более чем 10 градусов или смещен на более чем 20 см. Ключевой кадр и ключевые точки помещаются в очередь на первоочередную обработку.

Backend строит граф для представления геометрических отношений между положениями робота и ключевыми точками (ориентирами) [31], который обновляется каждый раз, когда Frontend создает новый ключевой кадр. В предлагаемой системе этот модуль включает в себя следующие этапы:

1. Граф взаимной видимости. Граф состоит из набора вершин ключевого кадра, набора ключевых точек и набора относительных ребер. Каждая вершина ключевого кадра сохраняет свое абсолютное положение, запоминая все видимые из данного положения ключевые точки, а также сохраняя соответствующие наблюдения. Ребро между двумя вершинами ключевых кадров содержит информацию о числе ключевых точек, которые видны из соответствующих вершин. В графе используется опорный ключевой кадр. Как только Frontend создает новый ключевой кадр, то он добавляется в граф.

---

**Algorithm:** Adaptive Motion Estimation.

---

**Data:**  $F_i, F_j$   
**Result:**  $T_{i,j}$ , inliers

```

1  $[S_i, S_j] = \text{ExtractAndMatch}(F_i, F_j);$ 
2  $[T_{i,j}, \text{inliers}, \text{err}] = \text{RANSAC}(S_i, S_j);$ 
3 if  $|\text{inliers}| \leq \mu V \text{err} \geq \tau_1$  then
4 |   if  $\text{err} \geq \tau_2$  then
5 | |    $[S_i, S_j] = \text{GetSubset}(S_i, S_j);$ 
6 | |    $T_{i,j} = \text{GICP}(S_i, S_j, I);$ 
7 |   else
8 | |    $T_{i,j} = \text{GICP}(S_i, S_j, T_{i,j});$ 
9 |   end
10 else
11 end

```

---

**Рис. 3.** Алгоритм адаптивной оценки движения.

2. Активная оптимизация окон. Активное окно состоит из ключевых кадров, связанных с текущим ключевым кадром и имеющих количество общих ключевых точек больше заданного количества (в нашем случае больше 100). Активное окно последнего ключевого кадра вставляется в граф и оптимизируется.
3. Замыкание цикла (Loop Closure). Для работы в большой окружающей среде требуется система, способная распознавать ранее посещенные места. Это помогает оптимизировать карту и траекторию движения, устранять накопленные ошибки и сохранять глобальную согласованность карты. Потенциальные кандидаты на оптимизацию выявляются с использованием описания ключевых кадров с помощью набора двоичных слов [32] и ранее обученного визуального словаря. Этот метод помогает найти потенциальные пересечения между текущим ключевым кадром и более старыми ключевыми кадрами. Замыкание цикла затем проверяется с помощью алгоритма адаптивной оценки движения с небольшой модификацией.
4. Глобальная оптимизация графа. При каждом успешном замыкании цикла к графу добавляется новое ограничение. Граф затем оптимизируется для распределения ошибки замыкания цикла по всему графу. Для построения трехмерной карты окружающей среды в предлагаемой системе используется древовидная структура данных — октодереву [33], которая приводит к компактному представлению и по своей природе допускает запросы к карте с разным разрешением [5].

### 3. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Работа предложенной системы (ASLAM) сравнивалась с работой с современных систем RGB-D SLAM на известной базе данных TUM RGBD [34]. Все эксперименты были проведены на ПК с процессором Intel Core i5-4200U с 8 ГБ оперативной памяти. Для оценки качества работы систем использовался критерий среднеквадратичное значение абсолютной ошибки траектории (ATE), т.е. отклонение оценки траектории от известной истинной траектории  $X$ . Ошибка вычисляется как

$$ATE(\hat{X}, X) = \sqrt{\frac{1}{n} \sum_{i=1}^n \left\| \text{trans}(\hat{X}_i) - \text{trans}(X_i) \right\|^2}. \quad (4)$$

Точность построения трехмерной карты окружающей среды для визуальных систем SLAM взаимосвязана с ATE ошибкой, то есть, чем точнее локализуется робот в окружающей среде, тем точнее была создана трехмерная карта этой среды и наоборот. Работа предложенной

системы сравнивалась с тремя современными системами: DVO-SLAM [35], RGBD-SLAM [5] и ORB-SLAM2 [36]. DVO-SLAM представляет собой прямую плотную систему, то есть модуль Frontend минимизирует фотометрическую ошибку, используя большую часть пикселей изображения для оценки положения камеры. RGBD-SLAM — это косвенная система, основанная на признаках, то есть она предварительно обрабатывает изображения для извлечения визуальных ключевых точек и нахождения соответствий с другими изображениями. Ее модуль Frontend оценивает положение камеры с использованием подхода RANSAC+ICP обычным способом. ORB-SLAM2 — это косвенная система, основанная на визуальных признаках ORB, ее модуль Frontend минимизирует расстояния между проекциями трехмерной модели и связанными точками на изображении для оценки положения камеры. В отличие от вышеупомянутых систем, предложенная система ASLAM использует только визуальные признаки FAST для адаптивного расчета траектории с применением подхода RANSAC+GICP. Эксперименты повторялись 30 раз, и средние значения результатов ошибок показаны в Таблице 2.

**Таблица 2.** Сравнение результатов работы тестируемых систем на разных сценах с точки зрения АТЕ(м).

Сцены	ASLAM	ORB-SLAM2	RGBD-SLAM	DVO-SLAM
fr1/desk	0.021	<b>0.016</b>	0.026	0.021
fr1/desk2	0.036	<b>0.022</b>	-	0.046
fr1/room	<b>0.036</b>	0.047	0.087	0.043
fr2/xyz	0.014	<b>0.004</b>	-	0.018
fr3/office	0.016	<b>0.010</b>	-	0.035

Отметим, что ORB-SLAM2 является лучшим алгоритмом в большинстве случаев, но обладает очень высокой вычислительной сложностью. Предложенный алгоритм лучше двух остальных тестируемых алгоритмов, близок по точности к ORB-SLAM2, а на некоторых сценах является лучшим. При этом предложенный алгоритм обладает минимальной вычислительной сложностью из всех тестируемых алгоритмов. На Рис. 4 показаны графы взаимной видимости для двух используемых сцен.



**Рис. 4.** Графы взаимной видимости для сцен: (а) fr1/room и (б) fr3/office.



## ЗАКЛЮЧЕНИЕ

В работе предложен адаптивный алгоритм визуального SLAM с использованием RGB-D камеры. Было показано, что комбинация FAST-BRIEF дает хорошие результаты в реальных приложениях, а адаптивный подход RANSAC+GICP при создании системы позволяет существенно улучшить работу визуального SLAM. Предложенная система на экспериментальных данных продемонстрировала, что способна построить трехмерную карту окружающей среды с высокой точностью и локализовать положение робота в реальном масштабе времени, используя только обычный персональный компьютер.

## СПИСОК ЛИТЕРАТУРЫ

1. Thrun S., Burgard, W., and Fox D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*., MIT Press, 2005.
2. Gonzalez-Fraga J., Kober V., and Gutierrez E. Accurate alignment of rgb-d frames for 3d map generation. *Proc. SPIE*, 2018, Vol. 10752, pp. 107522J-7.
3. Strasdat H., Montiel J. M. M., and Davison A. J. Visual SLAM: Why filter? *Image and Vision Computing*, 2012. Vol. 30, pp. 65–77.
4. Grisetti G., Kuemmerle R., Stachniss C., and Burgard W. A tutorial on graph-based SLAM. *Intelligent Transportation Systems Magazine*, 2010, IEEE Vol. 2, No. 4, pp. 31–43.
5. Endres F., Hess J., Sturm J., Cremers D., and Burgard W. 3D mapping with an RGB-D camera. *IEEE Trans. on Robotics*, 2014, Vol. 30, No. 1, pp. 177–185.
6. Fischler M. A. and Bolles R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981, Vol. 24, pp. 381–395.
7. Besl P. J. and McKay N. D. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1992, Vol. 14, pp. 239–256.
8. Makovetsii A., Voronin S., Kober V., and Voronin A. A point-to-plane registration algorithm for orthogonal transformations. *Proc. SPIE*, 2018, Vol. 10752, pp. 107522R-8..
9. Segal A., Hahnel D., and Thrun S. Generalized-ICP. *Conference: Robotics: Science and Systems V*, 2009, 10.15607/RSS.2009.V.021.
10. Miramontes-Jaramillo D., Kober V., and Diaz-Ramirez V. H., Karnaukhov V. Descriptor-Based Tracking Algorithm Using a Depth Camera. *Journal of Communications Technology and Electronics*, 2017, Vol. 62, No. 6, pp. 638–647.
11. Diaz-Escobar J., Kober V., Gonzalez-Fraga J.A. LUIFT: LUminance Invariant Feature Transform. *Mathematical Problems in Engineering*, 2018, Vol. 2018, Article ID 3758102.
12. Diaz-Escobar, J., Kober V., Karnaukhov V., Gonzalez-Fraga J.A. A new invariant to illumination feature descriptor for pattern recognition. *Journal of Communications Technology and Electronics*, 2018, Vol. 63, No. 12, pp. 1469–1474.
13. Diaz-Escobar J., and Kober V. A robust HOG-based descriptor for pattern recognition. *Proc. SPIE*, 2016, Vol. 9971, pp. 99712A-7.
14. Echeagaray-Patron B.A., Kober V., Karnaukhov V., Kuznetsov V. A method of face recognition using 3D facial surfaces. *Journal of Communications Technology and Electronics*, 2017, Vol. 62, No. 6, pp. 648–652.
15. Echeagaray-Patron B.A., Kober V. Face recognition based on matching of local features on 3D dynamic range sequences. *Proc. SPIE*, 2016, Vol. 9971, pp. 997131-6.
16. Shi J. and Tomasi C. *Good features to track Tech. rep.*, Ithaca, NY, USA, 1993.

17. Rosten E. and Drummond T. Machine learning for high-speed corner detection. *Proceedings of the 9th European Conference on Computer Vision*, 2006, Vol. Part I, ECCV'06, pp. 430–443, Springer-Verlag, Berlin, Heidelberg.
18. Rublee E., Rabaud V., Konolige K., and Bradski G. Orb: An efficient alternative to sift or surf. *Proceedings of the 2011 International Conference on Computer Vision*, 2011, Vol. ICCV '11, pp. 2564–2571, IEEE Computer Society, Washington, DC, USA.
19. Leutenegger S., Chli M., and Siegwart R. Y. BRISK: Binary robust invariant scalable keypoints. *Proceedings of the 2011 International Conference on Computer Vision*, 2011, Vol. ICCV '11, pp. 2548–2555, IEEE Computer Society, Washington, DC, USA.
20. Konolige K., Bowman J., Chen J., Mihelich P., Calonder M., Lepetit V., and Fua P. View-based maps. *The International Journal of Robotics Research*, 2010, Vol. 29, No. 8, pp. 941–957.
21. Calonder M., Lepetit V., Strecha C., and Fua P. Brief: Binary robust independent elementary features. *Proceedings of the 11th European Conference on Computer Vision*, 2010, Vol. Part IV, ECCV'10, pp. 778–792, Springer-Verlag, Berlin, Heidelberg.
22. Ortiz R. FREAK: Fast retina keypoint. *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, Vol. CVPR '12, pp. 510–517, IEEE Computer Society, Washington, DC, USA.
23. Levi G. and Hassner T. LATCH: Learned arrangements of three patch codes. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.
24. Lowe D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 2004, Vol. 60, pp. 91–110.
25. [25] Bay H., Ess A., Tuytelaars T., and Van Gool L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 2008, Vol. 110, pp. 346–359.
26. Garcia-Fidalgo E. and Ortiz A. Vision-based topological mapping and localization methods. *Robot. Auton. Syst.*, 2015, Vol. 64, pp. 1–20.
27. Mikolajczyk K. and Schmid C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, Vol. 27, pp. 1615–1630.
28. Mur-Artal R., Montiel J. M. M., and Tardós J. D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 2015, Vol. 31, No. 5, pp. 1147–1163.
29. Pire T., Fischer T., Castro G., De Cristoforis P., Civera J., and Jacobo Berles J. S-PTAM: Stereo Parallel Tracking and Mapping. *Robotics and Autonomous Systems (RAS)*, 2017, Vol. 93, pp. 27–42.
30. Paton M. and Kosecka J. Adaptive RGB-D localization. *Ninth Conference on Computer and Robot Vision*, 2012, pp. 24–31.
31. Strasdat H., Davison A. J., Montiel J. M. M., and Konolige K. Double window optimisation for constant time visual slam. *IEEE International Conference on Computer Vision*, 2011, pp. 2352–2359.
32. Galvez-Lopez D. and Tardós J. D. Bags of binary words for fast place recognition in image sequences. *Trans. Rob.*, 2012, Vol. 28, pp. 1188–1197.
33. Hornung A., Wurm K. M., Bennewitz M., Stachniss C., and Burgard W. OCTOMAP: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots*, 2013, Vol. 34, pp. 189–206.
34. Sturm J., Engelhard N., Endres F., Burgard W., and Cremers D. A benchmark for the evaluation of RGB-D SLAM systems. *International Conference on Intelligent Robot Systems (IROS)*, 2012.
35. Kerl C., Sturm J., and Cremers D. Robust odometry estimation for RGB-D cameras. *IEEE International Conference on Robotics and Automation*, 2013, pp. 3748–3754.
36. Mur-Artal R. and Tardós J. D. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 2017, Vol. 33, No. 5, pp. 1255–1262.

## Algorithm for the design of a three-dimensional map of the environment with a depth camera

**Ortiz-Gonzalez A., Kober V., Karnaukhov V., Mozerov M.**

A new algorithm for three-dimensional reconstruction of the surrounding environment is proposed. The algorithm is able to create accurate three-dimensional maps in real time using an inexpensive RGB-D camera. The algorithm can be used in autonomous mobile robotics, where the robot needs to localize itself in unknown environments by processing on-board sensors without external reference systems, such as a global positioning system. We analyze various combinations of the common detectors and descriptors of visual features in terms of their recognition efficiency. To match point clouds between consecutive frames, the iterative closest point (ICP) method is used. To improve the quality of the three-dimensional reconstruction, an adaptive approach to camera pose estimation is suggested. The proposed system is able to efficiently process complex scenes at high rate, and its performance on available benchmark datasets is comparable with that of the state-of-the-art systems.

**KEYWORDS:** three-dimensional reconstruction, iterative closest point method, depth camera, simultaneous localization and mapping (SLAM).