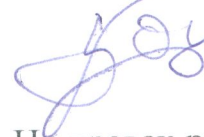


РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ



На правах рукописи

УДК 004.932.7, 004.4, 004.021

Боковой Андрей Валерьевич

**Исследование методов и разработка алгоритмов одновременного
картирования и локализации по видеопотоку единственной камеры**

Специальность 1.2.2 –

«Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель:

канд. физ.-мат. наук

Яковлев Константин Сергеевич

Москва – 2022

Оглавление

Оглавление	2
Введение	4
Глава 1. Анализ принципов работы современных методов и алгоритмов одновременного картирования и локализации по видеопотоку	12
1.1 Непрямые методы.....	14
1.2 Прямые методы	21
1.3 Описание коллекций для проведения полунатурного вычислительного эксперимента с методами vSLAM.....	26
1.4 Выводы по главе.....	39
Глава 2. Разработка оригинального алгоритма поиска замыканий траектории	42
2.1 Алгоритм поиска замыканий на траектории.....	45
2.2 Сопоставление изображений	47
2.3 Реализация алгоритма поиска замыканий на траектории в составе комплекса программной системы vSLAM	48
2.4 Результаты полунатурного эксперимента.....	50
2.5 Выводы по главе.....	58
Глава 3. Разработка вычислительно эффективных нейросетевых моделей для восстановления глубин изображений с последующим использованием в методах одновременного картирования и локализации.....	59
3.1 Разработанные нейросетевые модели	61
3.2 Алгоритмические оптимизации разработанных моделей искусственных нейронных сетей.....	65
3.3 Функции потерь.....	68

3.4 Исследование программной реализации разработанных моделей на встраиваемой платформе	70
3.5 Реализация в составе программного комплекса картирования и локализации	73
3.6 Выводы по главе	76
Глава 4. Разработка моделей для оценки результатов картирования и локализации, полученных методами mvSLAM	78
4.1 Оценка качества траектории	83
4.2 Разработка метода оценки качества картирования на основе модели выходных данных методов, решающих задачу mvSLAM	84
4.3 Экспериментальное исследование в симуляционной среде	89
4.4 Выводы по главе	97
Заключение	99
Список публикаций автора	103
Список литературы	106
Список рисунков	117
Список таблиц	120

Введение

Актуальность темы. Задача одновременного картирования и локализации (Simultaneous Localization and Mapping – SLAM) является одной из важнейших для обеспечения автономной навигации мобильных робототехнических систем. Ее решение позволяет строить карту местности и вычислять положение системы в каждый момент времени с использованием бортовых датчиков. Построенная карта и рассчитанная позиция, в свою очередь, позволяют мобильному роботу избегать столкновений с препятствиями, осуществлять планирование траектории до цели и др. При успешном решении задачи SLAM существенно расширяются возможности применения современных робототехнических систем в таких областях как поиск и спасение, мониторинг (местности, трубопроводов, объектов строительства и др.), ликвидация последствий природных и техногенных катастроф, сельское хозяйство (автоматическая уборка урожая) и др. Однако, в настоящий момент не существует универсального метода решения задачи SLAM, подходящего, в том числе, для применения на малых робототехнических системах, которые не могут быть оснащены широким набором датчиков и мощным вычислителем в силу технических и экономических ограничений. Именно на решение этой проблемы и направленно исследование, что обуславливает его актуальность.

Степень разработанности темы. Обычно задача SLAM рассматривается для конкретных областей применения и для каждого типа датчиков отдельно. Например, существуют разные подходы для сканирующих лазерных дальномеров, стереокамер, сонаров и других сенсоров, а также для наземных, летающих, подводных робототехнических устройств. Одной из наиболее сложных постановок задачи SLAM является монокулярный визуальный SLAM (monocular vision-based SLAM – mvSLAM), когда используются только данные с единственной камеры. Подобная формулировка задачи обычно возникает, когда речь идет о мобильных роботах, которые в силу ограниченной грузоподъемности и емкости батареи не

могут быть оснащены большим количеством сенсоров. Чаще всего, монокулярная камера является единственным сенсором на борту.

К распространенным методам, решающим задачу mvSLAM, относятся MonoSLAM, PTAM, ORB-SLAM/ORB-SLAM2, DSO, LSD-SLAM, DTAM, SVO, CNN-SLAM. Среди исследователей, занимающихся созданием методов, алгоритмов и моделей, который относятся к задаче mvSLAM, можно выделить A.J. Davison, I.D. Reid, G. Klein, D. Murray, R. Mur-Atal, J.M.M. Moontiel, R. Wang, M. Schworer, D. Cremers, J. Engel, T. Schops, R.A. Newcombe, S.J. Lovegrove, C. Forrester, M. Pizzoli, D. Scaramuzza, K. Tateno, F. Tombari, I. Laina, N. Navab, Е.Н. Соснова, С.Л. Зенкевича, В.С. Лемпицкого и В.И. Кобера.

Однако, несмотря на повышенное внимание исследователей, известные в настоящий момент методы решения задачи mvSLAM обладают рядом недостатков, среди которых стоит упомянуть низкую точность построения карт и их малую детализацию, низкую эффективность алгоритмов поиска замыканий траектории и ряд других. Отдельно стоит отметить тот факт, что для работы большинства методов, решающих задачу mvSLAM в режиме реального времени, требуются мощные вычислители, что затрудняет их применение на борту мобильных роботов. Таким образом, представляется целесообразным исследование и разработка методов решения задачи одновременного картирования и локализации по видеопотоку единственной камеры, лишенных (части) указанных недостатков, что определяет цель и задачи работы.

Целью исследования является повышение степени автономности малых робототехнических систем посредством разработки и улучшения методов, решающих задачу одновременного картирования и локализации по видеопотоку единственной камеры.

Для достижения поставленной цели были сформулированы и решены следующие **задачи**:

1. Разработать эвристический алгоритм обнаружения замыкания траектории для повышения точности и вычислительной эффективности методов решения задачи картирования и локализации по видеопотоку.
2. Разработать модель искусственной нейронной сети, позволяющей решать задачу восстановления глубины изображений в режиме реального времени с приемлемым качеством для последующего использования в задаче одновременного картирования и локализации по видеопотоку единственной камеры.
3. Построить модель оценки качества карт (трехмерных облаков точек), получаемых с помощью методов решения задачи mvSLAM, с учетом особенностей задачи одновременного картирования и локализации.
4. Провести экспериментальные исследования разработанных алгоритмов и моделей на открытых коллекциях данных, в симуляционных средах и на реальных робототехнических системах.

Методология и методы исследования основаны на теории вероятности, математической статистике, линейной алгебре, аналитической геометрии, теории графов, цифровой обработке сигналов, компьютерном зрении, методах разработки и тестирования программного обеспечения для ЭВМ.

Научная новизна работы состоит в следующем.

Предложен новый алгоритм обнаружения замыканий траекторий, основанный на геометрических ограничениях движения робототехнической системы. В отличие от имеющихся аналогов предложенный алгоритм использует информацию о пройденной робототехнической системой траектории для эвристического ограничения области поиска замыкания. Это позволяет существенно повысить вычислительную эффективность метода решения задачи mvSLAM, а также оказывает положительное влияние на качество построения траектории (точность построения траектории увеличивается).

Разработана оригинальная модель искусственной нейронной сети с использованием ускоренной (на графическом процессоре) операции совмещения

матриц для предсказания карт глубин изображений. Были предложены функции потерь, уменьшающие ошибку предсказания на малых расстояниях. Полученная модель позволила решить задачу mvSLAM в режиме реального времени на бортовом вычислителе малой робототехнической системы. Проведенные исследования качества предсказания глубины и скорости работы на открытых коллекциях данных и реальной робототехнической системы показали сопоставимые по качеству результаты с современными моделями при увеличенной производительности.

Предложена новая модель оценки качества карт (трехмерных облаков точек), получаемых методами решения задачи mvSLAM. В отличие от аналогов, опирающихся на сопоставление точек по принципу ближайшего соседа, предложенная модель основана на сопоставлении точек двух множеств (эталонного и построенного) с учетом позиций камеры, которые были вычислены в ходе реконструкции трехмерной карты из двумерного изображения. Разработана новая коллекция данных для проведения полунатурного эксперимента в симуляционной среде. Проведены исследования известных алгоритмов, решающих задачу mvSLAM с помощью полученных модели оценки и коллекции на разных входных данных. Показано преимущество разработанного подхода к оценке качества картирования перед стандартными.

Теоретическая и практическая значимость работы. Теоретическая значимость работы обуславливается комплексом разработанных алгоритмов и моделей, которые создают основу как для построения новых методов решения задачи mvSLAM, так и для улучшения существующих. Полученная в ходе работы модель оценки карт может быть применена для исследования качества картирования методов, решающих задачу mvSLAM. Практическая значимость работы заключается в реализации разработанного алгоритма и моделей в виде комплекса программных средств для реальных робототехнических систем (как наземных, так и воздушных). Полученные в ходе работы теоретические и

практические результаты могут быть использованы для повышения степени автономности робототехнических систем различного типа и назначения.

Достоверность полученных результатов подтверждается согласованностью разработанных алгоритмов и математических моделей с экспериментальными результатами, полученными на открытых коллекциях данных (NYU Dataset v2, Malaga Dataset, TUM RGB-D Dataset и др.) и реальных робототехнических системах, успешной апробацией результатов на научных конференциях и реализацией полученных моделей и алгоритма для реальной робототехнической платформы.

Апробация работы. Основные результаты докладывались на следующих семинарах и конференциях: XIX Национальная конференция по искусственному интеллекту с международным участием (КИИ-2021), International Siberian Conference on Control and Communications (SIBCON-2021), XVI международная конференция по электромеханике и робототехнике «Завалишинские чтения 2020», The 9th European Conference on Mobile Robots (ECMR 2019), Пятый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2019), The 3rd International Conference on Interactive Collaborative Robotics (ICR 2018), The 2018 International Conference on Artificial Life and Robotics (ICAROB2018), The 6th International Conference on Analysis of Images, Social networks and Texts, (AIST 2017) и ряде других профильных конференций и семинаров.

Основные результаты диссертации были получены в процессе выполнения работ по гранту №075-15-2020-799 «Методы построения и моделирования сложных систем на основе интеллектуальных и суперкомпьютерных технологий, направленные на преодоление больших вызовов» Министерства науки и образования РФ, а также по следующим грантам РФФИ: № 15-07-07483, «Исследование методов и разработка алгоритмов картирования, локализации и автоматического планирования траектории сложных технических объектов, обладающих многими степенями свободы»; № 17-29-07053, «Гетерогенные

иерархические системы динамического планирования и управления поведением интеллектуального агента»; № 18-37-20032, «Методы управления автономными техническими объектами на основе планирования траектории в среде с динамическими препятствиями».

Личный вклад. Все положения, выносимые на защиту, изложенные в диссертации, принадлежат лично автору. В совместных работах автор принимал непосредственное участие в выборе направления и задач исследования, в программной реализации и обсуждении результатов экспериментальных исследований.

Публикации. Основные результаты по теме диссертации изложены в 17 публикациях, в том числе: 2 изданы в журналах, рекомендованных ВАК, 9 - в изданиях, входящих в международные базы цитирования Scopus и Web of Science.

Объем и структура работы. Диссертация состоит из введения, трех глав и заключения. Полный объем диссертации составляет 120 страниц, включая 35 рисунков и 9 таблиц. Список литературы содержит 110 наименований.

Основные положения, выносимые на защиту:

1. Предложен алгоритм обнаружения замыканий траекторий на основе эвристического ограничения области поиска визуального сходства изображений, позволяющий повысить точность локализации и вычислительную эффективность работы методов, решающих задачу одновременного картирования и локализации по видеопотоку единственной камеры.
2. Разработана модель искусственной нейронной сети для восстановления глубины изображений на основе автокодировщика с ускорением отдельных операций с помощью алгоритма совмещения матриц на графическом процессоре, позволяющая значительно повысить производительность восстановления глубины изображений при сохранении высокой точности восстановления, достаточной для

эффективного решения задачи одновременного картирования и локализации по видеопотоку единственной камеры.

3. Разработана модель оценки качества карт, получаемых методами решения задачи одновременного картирования и локализации по видеопотоку единственной камеры, – трехмерных облаков точек, – основанная на сопоставлении точек двух множеств (эталонного и построенного) с учетом позиций камеры, которые были вычислены в ходе реконструкции трехмерной карты из двумерного изображения.

Краткое содержание глав. Первая глава посвящена анализу современных методов, решающих задачу одновременного картирования и локализации по видеопотоку, а также современных методов для обработки карт и восстановления глубины изображений. Проанализированы их преимущества и недостатки. На основе проведенного анализа, поставлены задачи научно-исследовательской работы.

Вторая глава посвящена разработке алгоритма обнаружения замыканий траектории в составе известных методов, решающих задачу одновременного картирования и локализации по видеопотоку. Проводится анализ стандартных решений в предметной области на примере открытой реализации методов ORB-SLAM2 и LSD-SLAM. Учитывая выявленные недостатки стандартных алгоритмов замыкания траектории, был предложен эвристический алгоритм ограничения области поиска замыканий траектории на основе траектории движения робота. Проводится экспериментальный анализ предложенного алгоритма и показывается его преимущество перед оригинальными алгоритмами поиска замыканий траектории.

В третьей главе приводится описание задачи восстановления глубины изображений с помощью искусственных нейронных сетей, в контексте задачи картирования и локализации по видеопотоку на встраиваемой системе Nvidia Jetson TX2 в реальном времени. Проведен анализ современных моделей искусственных нейронных сетей, решающих поставленную задачу, и предложен ряд собственных

архитектур, функций потерь и отдельных блоков, которые уменьшают время обработки моделью одного изображения, при этом не теряя точности восстановления глубины по сравнению с современными архитектурами. Проведен экспериментальный анализ, который показал превосходство разработанного решения и применимость полученных архитектур для картирования и локализации робототехнического устройства на базе встраиваемой системы Nvidia Jetson TX2 в реальном времени.

В четвертой главе описана проблема оценки качества картирования методов, решающих задачу mvSLAM. Описана разработанная модель оценки качества карт, основанная на сопоставлении точек двух множеств с учетом позиций камеры, которые были вычислены в ходе реконструкции трехмерной карты. Разработана коллекция данных для исследования методов картирования и локализации, основанная на фотореалистичном симуляторе. Приведены результаты экспериментального исследования известных методов с использованием разработанных моделей.

Глава 1. Анализ принципов работы современных методов и алгоритмов одновременного картирования и локализации по видеопотоку

Существует большое количество методов, позволяющих определять текущее местоположение беспилотного летательного аппарата, но, как правило, они используют информацию от спутниковых навигационных систем (GPS/ГЛОНАСС), которая доступна только на открытой местности. Для навигации в закрытых помещениях необходимо полагаться на ИНС, оптические и другие датчики.

Одним из активно развивающихся направлений исследований в настоящее время является навигация по данным, полученным с оптических датчиков. Одной из задач этого направления является разработка методов и алгоритмов, решающих задачу одновременного картирования и локализации по видеопотоку, полученному с единственной камеры (monocular vision-based Simultaneous Localization and Mapping – mvSLAM) [43,67,57]).

Основной задачей методов, решающих задачу mvSLAM, является построение трехмерной карты неизвестной местности (как правило, в виде трехмерного облака точек (pointcloud) [102] или двумерной сетки [49]) и вычисление текущей позиции в построенной карте по видеопотоку. Построенная модель местности может быть использована в дальнейшем в задачах планирования траектории, автономной навигации, при реактивном движении сложных робототехнических систем.

Визуальное одновременное картирование и локализация (vSLAM) является одной из самых сложных задач в области компьютерного зрения и робототехники. Методы, решающие задачу vSLAM, основанные только на информации, полученной от минимального набора миниатюрных датчиков (монокулярная или стереокамера, инерциальная навигационная система), лежат в основе навигации мобильных роботов. Особенно они представляют большую ценность для компактных беспилотных летательных аппаратов (которые по умолчанию не могут

быть оснащены габаритными, мощными датчиками). Недавно был достигнут заметный прогресс в области методов vSLAM для беспилотных летательных аппаратов (БПЛА), (см. работы [63,89]). Однако все еще существует большой набор реальных проблем и сценариев, которые не могут быть успешно решены существующими методами на основе видеопотока. Основными причины – следующие:

Во-первых, это время обработки изображения. Современные встроенные компьютеры [12,13], которые могут быть установлены на компактных беспилотных летательных аппаратах, не обладают достаточным быстродействием для выполнения типичных алгоритмов в режиме реального времени. Использование внешних источников для удаленных вычислений не всегда является решением, поскольку оно снижает мобильность (роботизированная система вынуждена постоянно обмениваться огромным объемом информации с удаленной станцией управления, используя проводной или беспроводной канал) и препятствует тому, чтобы роботизированная система была полностью автономной.

Во-вторых, плохое качество изображений [59,44]. Небольшие камеры, которые обычно устанавливают на малые БПЛА, сильно подвержены влиянию условий окружающей среды (свет, погода и т.д.) и часто производят видеопоток, содержащий многочисленные искажения, шумы и другие артефакты. Необходимо применять различные методы фильтрации [39,91] для предварительной обработки видеопотока и повышать эффективность методов, решающих задачу mvSLAM.

Качество, скорость и точность картирования и локализации по видеопотоку зависит от многих факторов, начиная от траектории движения камеры (при поворотах без перемещения резко падает качество локализации, при повороте на месте карта восстанавливается с сильными искажениями), заканчивая условиями, при которых работает алгоритм (освещение, качество камеры, шумы на изображениях и т.д.). В настоящее время, не существует универсального метода, который позволяет полностью решить vSLAM задачу для всех условий работы (окружающей среды, мощности вычислителей и т.д.), поэтому современные

методы разделяются на классы. Каждый класс лучше или хуже подходит под конкретные ограничения, которые налагаются в процессе эксплуатации робототехнической системы. Методы разделяются на:

1. Непрямые методы (indirect methods) [92], которые быстро работают на современных вычислителях и устойчивы к искажениям на изображениях, однако строят малодетализированные карты.
2. Прямые методы (direct methods) [84], которые работают медленно, но позволяют строить высокодетализированные карты.

Разберем каждый класс подробнее.

1.1 Непрямые методы

Принцип работы не прямых методов основывается на использовании части информации с последовательности изображений для построения карты и восстановления траектории движения камеры. Как правило, в качестве базовых элементов используются либо особенности изображений (ключевые точки, углы, прямые и т.д.) [53,97], либо контрастные области на изображениях [17].



Рис. 1. Пример карты, построенной непрямым методом ORB-SLAM2.

Использование единственной камеры приводит к ряду проблем, которые необходимо рассматривать как отдельные подзадачи, а именно:

- выделение областей или особенностей (особых точек, углов и т.д.) изображений, которые будут использоваться как элемент построенной модели местности (карты);
- построение модели неизвестной местности по двум или более кадрам видеопотока [90,51];
- локализация системы в построенной модели [3,81,98];
- построение и корректировка пройденной траектории [40,10].

В связи с большими объемами данных, поступающих с камеры, дополнительным условием (при решении указанных выше задач) является возможность обработки поступающих изображений, локализации и построения карты в режиме реального времени на современных бортовых вычислителях. Невыполнение этого условия может стать причиной потери актуальности данных и, следовательно, невозможности навигации в неизвестной среде.

Выделение областей и особенностей изображений. Для построения модели неизвестной местности и локализации БПЛА в ней необходимо выделить устойчивые признаки (особенности) изображений, которые будут являться элементами будущей карты. Определяющим условием при выделении особенностей является возможность сравнивать полученные признаки для разных изображений и находить однозначное соответствие между ними [66,71,60]. В основном под особенностями понимаются точки локального градиента яркости и дескрипторы, которые представляют из себя вектор, элементами которого являются градиенты в некоторой области вокруг полученной особенности.

Распространенные методы выделения особенностей изображения, такие как SIFT [62] и SURF [4], используемые в MonoSLAM, позволяют получать инвариантные повороту, масштабированию и (в некоторой степени)

перспективным искажениям дескрипторы размером 64 или 128 элементов, которые соответствуют каждой выделенной особенности. Модель местности в таком случае модель местности (карта) представляет из себя облако точек, элементами которой и являются эти особенности.

Метод FAST-10 [77], используемый в алгоритме PTAM, выделяет углы в качестве особенностей изображения и, соответственно, вычисляет дескриптор каждого угла. На Рис. 2 показаны особенности изображений, выделенные алгоритмами SIFT, SURF и FAST-10.

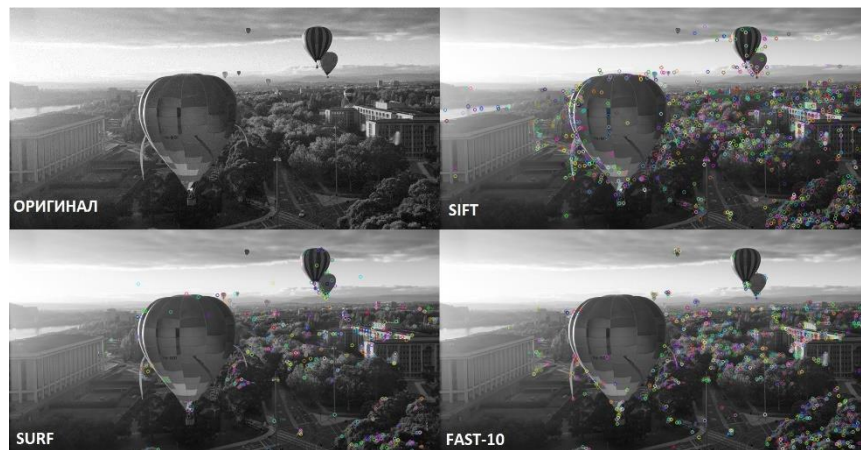


Рис. 2. Примеры выделения особенностей с помощью методов SIFT, SURF и FAST-10

Главными преимуществами вышеописанных методов выделения особенностей является скорость обработки поступающих изображений и составления дескрипторов полученных особенностей, позволяющая выполнять эти алгоритмы в реальном времени даже на слабых бортовых вычислителях и использовать их как составной элемент алгоритмов mvSLAM.

Недостатком использования алгоритмов SIFT, SURF и FAST-10, является скорость сопоставления дескрипторов разных изображений видеопотока. По причине того, что на изображении могут быть найдены тысячи особенностей, каждой из которых соответствует дескриптор размером от 64-х элементов, то их полный перебор и сравнение всех дескрипторов может занимать длительное время.

Для сокращения времени сопоставления наборов дескрипторов между собой используется представление дескрипторов в виде К-мерного дерева и, соответственно, применяются алгоритмы поиска на К-мерных деревьях.

В качестве входных данных для составления модели местности могут использоваться наиболее контрастные области (а не отдельные точки) изображения. Подход впервые был представлен авторами метода LSD-SLAM (см. Рис. 3). Он позволяет строить детализированные модели, а алгоритм, его реализующий, может выполняться в реальном времени без использования ускорения на GPU.



Рис. 3. Карта, построенная с использованием контрастных областей изображений (LSD-SLAM Foodcourt Dataset). Изображение из [26].

Построение модели местности и локализация. Для построения модели местности (карты) с помощью единственной камеры используются алгоритмы, позволяющие находить положение определенной точки в пространстве, используя 2 кадра, на которых эта точка видна. В качестве точек карты используются вышеописанные особенности изображений.

Для начала необходимо откалибровать камеру [106]. Это позволит найти внутренние параметры камеры: фокусные расстояния, углы наклона пикселей и принципиальную точку. Эти параметры используются как для устранения искажений на изображениях, так и для составления матрицы внешней калибровки и построения частичной карты глубины изображения (матрицы, у которой

значения, соответствующие определенным пикселям, равны расстояниям от камеры до проецируемого предмета) методом триангуляции с использованием соседних изображений. Пример полной карты глубины (расстояния рассчитаны для каждого пикселя) представлен на Рис. 4.

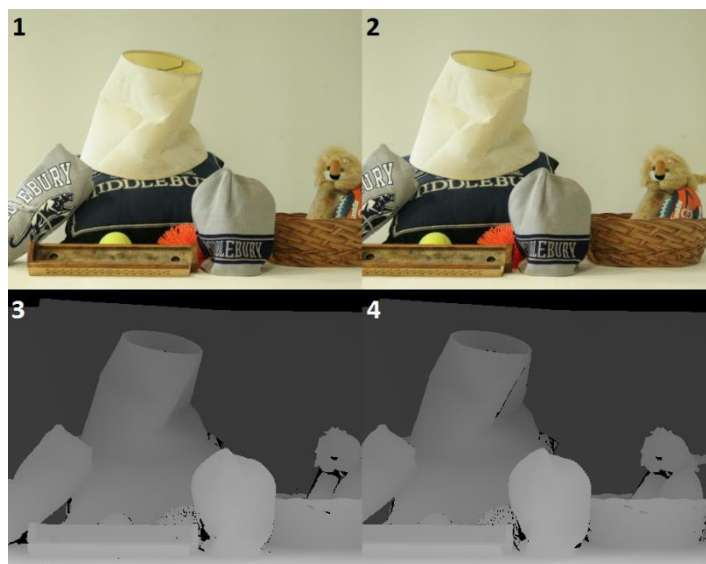


Рис. 4. Карта глубины, составленная по паре изображений (Middlebury Stereo Datasets). 1 – изображение с левой камеры, 2 – изображение с правой камеры, 3, 4 – соответствующие им карты глубины. Чем темнее пиксель на карте глубины – тем дальше объект.

Зная параметры калибровки камеры, можно найти фундаментальную матрицу и существенную матрицу. Фундаментальная матрица, в данном случае, необходима для вычисления уравнений эпиполярных линий, которые необходимы для восстановления позиции камеры. Для вычисления фундаментальной матрицы обычно используется Eight-point algorithm [37] и его вариации, которые позволяют найти фундаментальную матрицу по 8-ми или более сопоставленным точкам на двух изображениях. Существенная матрица позволяет восстановить положение и поворот камеры относительно ее предыдущего состояния по двум изображениям.

Для нахождения матрицы достаточно знать внутренние параметры камеры, полученные на этапе калибровки камеры.

Процесс вычисления расстояний до особых точек и нахождения положения камеры происходит в каждый для каждого вновь поступившего изображения с камеры. Текущий снимок сравнивается с предыдущим. Данные о положении камеры поступают на вход расширенному фильтру Калмана (Extended Kalman Filter) [46], где вычисляется текущее положение относительно построенной карты и производится корректировка предыдущих положений согласно полученным данным.

Вышеописанные вычисления производятся для каждого вновь поступившего изображения с камеры, что накладывает требования к скорости выполнения каждого алгоритма. В совокупности все алгоритмы должны выполняться до поступления нового изображения с камеры. Иначе возможна потеря актуальности данных и, следовательно, затрудненность или невозможность навигации БПЛА или иного робототехнического устройства.

На данный момент существуют программные реализации описанных алгоритмов, выполняющихся в режиме реального времени для видеоряда с частотой более 30-ти кадров в секунду. Наличие открытых реализаций позволяет оценить характеристики основных не прямых методов.

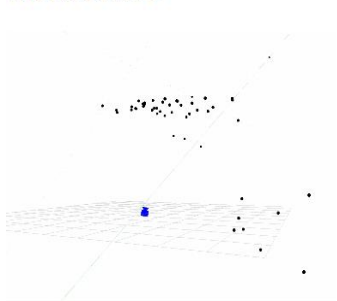
Характеристики не прямых алгоритмов. Для оценки характеристики не прямых методов была сформирована коллекция, состоящая из 63 видеозаписей помещений коридорного типа. Были смоделированы движения БПЛА, оснащенного камерой, а именно движения вперед/назад, изменение высоты, повороты вокруг своей оси и т.д. Все изображения видеопотока имеют разрешение 640x480 pix. Частота кадров – 30 кадр./с. Камера была откалибрована алгоритмами открытой библиотеки компьютерного зрения OpenCV [9] с помощью паттерна типа “шахматная доска” размером 9x6.

Платформа, на который производился эксперимент, представляет собой персональный компьютер со следующими основными характеристиками: CPU – Intel Core i7 3770K 3,4 GHz, 16 GB RAM, ОС – Linux

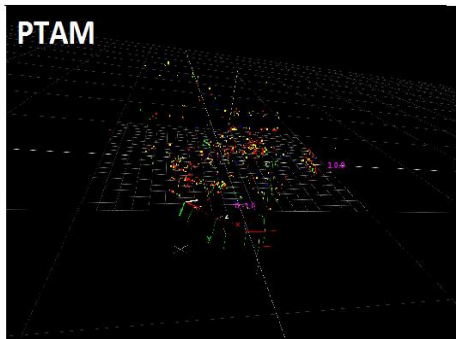
Для тестирования были выбраны свободные реализации алгоритмов MonoSLAM, PTAM и LSD-SLAM на языке C++ для фреймворка Robot Operating System (ROS). Исходные коды методов скомпилированы с использованием набора компиляторов GCC 4.9.2 без использования возможности ускорения параллельных вычислений (таких, как OpenCL или CUDA).

Параметрами для сравнения являлись время обработки одного изображения (пре-процессинг + выделение особенностей), время обновления карты и текущего положения системы, количество устойчивых особенностей (данный параметр также характеризует детализацию карты) и среднее отклонение от действительной траектории. Примеры карт, построенных с помощью каждого из методов представлены на Рис. 5.

MonoSLAM



PTAM



LSD-SLAM

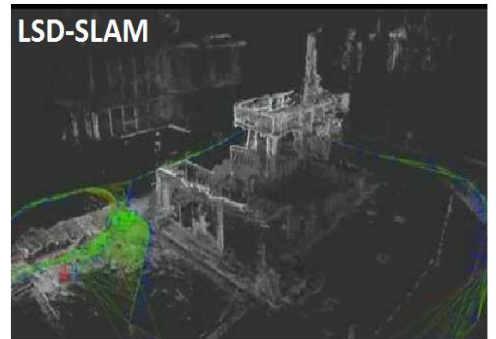


Рис. 5. Примеры карт, построенных с помощью методов MonoSLAM, PTAM и LSD-SLAM

Результаты эксперимента приведены в Таблица 1.

Таблица 1. Результаты экспериментального исследования не прямых методов vSLAM.

Метод	Время обработки кадра (мс)	Время обновления карты (мс)	Кол-во устойчивых признаков	Среднее отклонение от траектории (см)
MonoSLAM	8.35	4.7	473	8.1
PTAM	14.7	32.95	5837	7
LSD_SLAM	12.4	20	<20 000	3.8

По этим данным можно заключить, что метод MonoSLAM является самым быстрым из 3-х представленных методов, однако обладает наименьшей детализацией карты и наибольшим отклонением построенной траектории от действительной. Метод PTAM – самым медленным, но обладающим большей детализацией карты, нежели MonoSLAM и более высокой точностью построенной траектории. Метод LSD-SLAM обладает средними среди этих методов характеристиками по скорости выполнения алгоритма, но зато позволяет строить высокодетализированные карты и его среднее отклонение от заданной траектории – наименьшее.

1.2 Прямые методы

В отличие от непрямых методов, прямые методы [34] используют всю информацию на изображениях для восстановления карты глубины и построения карты местности, а не абстракцию в виде особенностей изображений. Эта группа методов позволяет строить высокодетализированные и точные карты, но отличается медленной обработкой каждого кадра (без ускорения на графическом процессоре). Однако, искажения, которые могут присутствовать на изображениях могут существенно снизить качество построенной карты.

Например, в методе DTAM [70] локализация происходит путем сравнения текущего изображения с синтетическими изображениями, полученными из трехмерной карты. Т.е., по сути решается задача сопоставления 2D изображения с 3D средой. Картирование происходит с использованием метода multi-baseline stereo

[42], а затем выполняется оптимизация карты методом из [108]. Инициализация выполняется с помощью методов эпиполярной геометрии, как в алгоритме PTAM. Несмотря на то, что метод DTAM оптимизирован для работы в реальном времени, быстродействия встраиваемых систем для обработки видеопотока будет недостаточно, что ограничивает применение метода в задаче автономной навигации робототехнических систем.

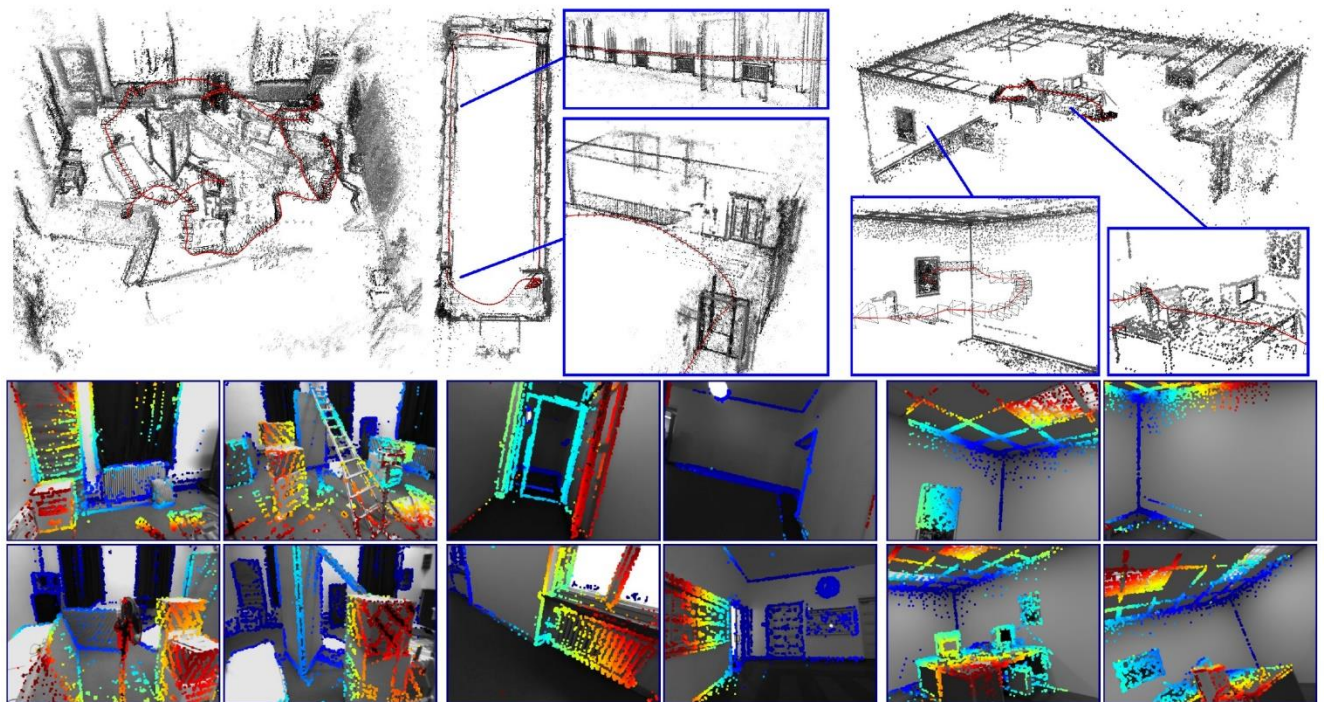


Рис. 6. Пример работы прямых методов. Изображение из [25]

Методы SVO [27] и DSO [25] используют особенности изображений для локализации, что сказывается на быстродействии в положительную сторону. Картирование, в свою очередь, происходит с использованием всей информации на изображениях. Для корректировки накапливающейся ошибки в этих методах используются различные геометрические и фотометрические фильтры, которые позволяют повысить качество входящих изображений.

В методе DSO, к примеру, изображение разбивается на несколько блоков, а затем точки с наибольшей интенсивностью яркости используются для

восстановления инвертированной карты глубины, которая затем используется для картирования и локализации.

К прямым методам можно отнести RGB-D vSLAM [82] (картирование и локализация в таких методах осуществляется с помощью датчиков глубины, т.е. карта глубины измеряется на аппаратном уровне), в которых карта глубины восстанавливается нейросетевыми моделями.

Применение нейросетевых моделей в задаче vSLAM. Для решения задачи одновременного картирования и локализации по видеопотоку в реальном времени на малых беспилотных летательных аппаратах, как правило, используются непрямые методы. Детализированность карт, полученных этими методами, может быть недостаточной для задач автономной навигации БПЛА, в частности, для задачи планирования траекторий [109,7,31].

Проблемы низкой детализированности карт, а также масштабирования и устойчивости работы при повороте камеры без движения, можно решить, если для восстановления глубины изображения, полученного с единственной камеры, использовать не сравнение особенностей/пикселей/контрастных областей с предыдущими изображениями, а использовать сверточные нейронные сети. Восстановление глубины из единственного изображения позволит добавить к каждому изображению на видеопотоке соответствующую карту глубины, т.е. в качестве входных данных для алгоритма mvSLAM будут поступать RGB-D изображения.

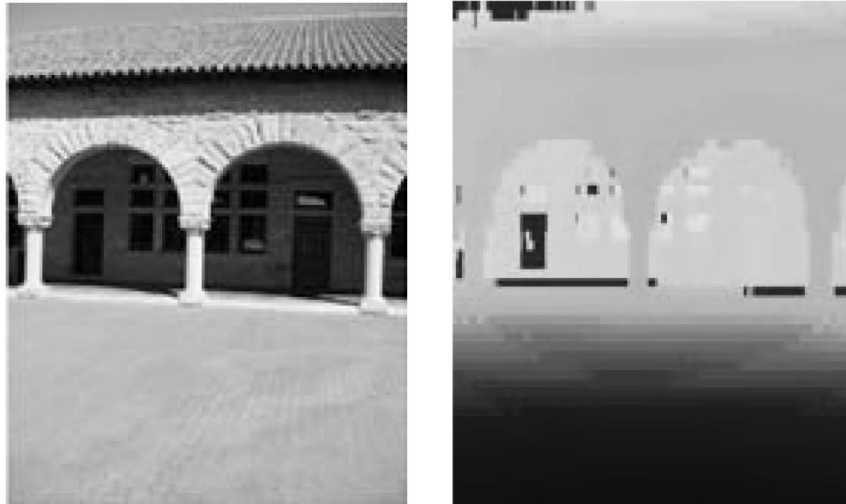


Рис. 7. Исходное изображение (слева) и восстановленная карта глубины (справа).

Яркость пикселей на изображении справа означают расстояния до соответствующих объектов (ярче - дальше).

Дополнив RGB изображения с камеры соответствующей картой глубины, можно рассматривать поставленную задачу одновременного картирования и локализации по видеопотоку, полученному с единственной камеры как задачу RGB-D vSLAM [48]. Использование RGB-D изображений позволит существенно повысить детализированность карт, без потери точности и времени, по сравнению с прямыми методами [68]. Озвученный подход уже предлагался в работах, посвященных реконструкции карты глубины по единственному изображению, семантической разметке и нахождении замыкания траекторий [29,23]. Но существующие на данный момент работы в области vSLAM с использованием нейронных сетей [18,45,47,104,73] либо не предоставляют возможности повторить результаты и проверить их в реальных условиях эксплуатации робототехнических систем, либо даже при наличии исходного кода не позволяют обрабатывать кадры в реальном времени и производить картирование и локализацию с использованием таких кадров, поскольку либо целевыми платформами являются либо мощные рабочие станции, либо основной задачей при разработке архитектур являлось максимальное качество без учета быстродействия. Поэтому возникла

необходимость разработки оригинальной архитектуры искусственной нейронной сети, которая позволяла бы работать со сравнимым с современными решениями качеством в реальном времени.

Для разработки собственной модели сверточной нейронной сети, необходимо понять по какому принципу строятся современные искусственные нейронные сети для задачи восстановления глубины изображений. В общем случае, сверточная нейронная сеть представляет собой набор скрытых слоев: слой свертки (convolutional layer), слой субдискретизации (pooling layer) и полносвязной нейронной сетью (fully connected network).

Слой свертки используется для выделения карт особенностей изображений (feature maps) посредством применения операции свертки матриц, веса которых вычисляются во время обучения, и исходного изображения. В результате операции, получается набор карт особенностей.

Слой субдискретизации уменьшает размер карты особенностей, полученной на шаге свертки. Уменьшение размера обусловлено тем, что если на шаге свертки уже был получен набор особенностей изображения, то уменьшение карт изображения не даст потери информации для последующих слоев. Операции субдискретизации позволяет существенно снизить объем вычислений и памяти, необходимой для обработки карт особенностей.

Полносвязная нейронная сеть получается в результате объединения карт признаков и преобразования их в вектор. Сеть представляет собой некий набор абстрактных параметров изображения.

Описанная выше архитектура является стереотипной для задачи классификации изображений, но для задачи выделения глубины из изображения, полученной полносвязной нейронной сети будет недостаточно. Необходима модификация общей архитектуры сверточной нейронной сети путем добавления дополнительных операций развертки (deconvolution) и повышения дискретизации (upsampling), обратных тем, что применяются на этапах свертки и субдискретизации [38].

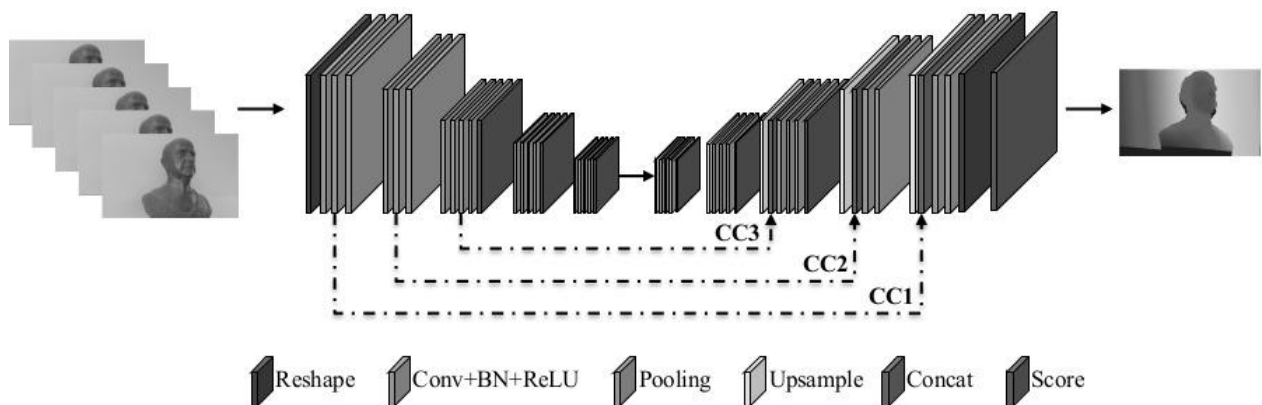


Рис. 8. Архитектура сети Deep Depth From Focus Net (DDFFNet).

В результате, на выходе искусственной нейронной сети получается не набор абстрактных признаков, а карта весов, которую можно интерпретировать как карту глубины. Изображение и восстановленную из него карту глубины можно использовать как входные данные для алгоритмов mvSLAM.

Для обучения сети может быть применен подход обучения с учителем (supervised learning) с использованием набора открытых данных, полученных как с единственных камер, так и с RGB-D датчиков. Основным условием является наличие изображения и соответствующей ему карты глубины или возможность эту карту глубины извлечь из карты местности (путем сопоставления позиции камеры и карты). Сам процесс обучения будет проходить на рабочей станции для сокращения времени на обработку изображений. В качестве основных коллекций могут быть использованы SUN RGB-D [85], ICL-NUIM RGB-D Benchmark Dataset [36], TUM RGB-D SLAM Dataset, Malaga Dataset 2009 и др.

1.3 Описание коллекций для проведения полунатурного вычислительного эксперимента с методами vSLAM

В задачах одновременного картирования и локализации по видеопотоку возникает потребность в оценке критериев разработанных методов и алгоритмов, а

также необходимость в сравнении полученных методов с уже известными. Параметрами для оценки могут являться: скорость работы алгоритма, уровень детализации построенной карты, среднее отклонение полученной траектории от реальной и т.д. Для оценки этих критериев используются специальные коллекции данных.

Как правило, эти коллекции представляют из себя набор видеозаписей, снятых в определенной местности с использованием одной камеры или стереопары. Каждый набор таких видеозаписей рассчитан на запуск программных реализаций алгоритмов одновременного картирования и локализации по видеопотоку с использованием единственной камеры, стереопары или RGB-D датчиков. В зависимости от того, на какой вид методов рассчитана коллекция различаются и форматы данных.

Для тестирования методов одновременного картирования и локализации по видеопотоку с использованием единственной камеры могут быть использованы цифровые видеозаписи известной или неизвестной местности. Камеры, с которых производится видеозаписи, варьируются от дешевых любительских камер, оснащенных КМОП-сенсором, не имеющих возможности настройки экспозиции, визуальными дефектами видеоряда типа Rolling shutter и заканчивая промышленными камерами оснащенными ПЗС-датчиками, имеющими широкие настройки композиции, возможностью съемки видео с количеством кадров в секунду более 30.

Использование камер разных классов обуславливается прикладными задачами алгоритмов одновременного картирования и локализации. Например, если планируется использование алгоритмов vSLAM (visual-based Simultaneous Localization and Mapping) на робототехнических устройствах малого типа, автономная работа которых зависит от емкости батареи, установленной на это устройство, а также в силу малых габаритов использование промышленных камер не представляется возможным. В таком случае использование камер с КМОП-датчиком является одним из выходов в силу малого энергопотребления, малых

размеров и дешевизны таких камер (что дает возможность быстрой замены камерного модуля в случае выхода его из строя).

При использовании КМОП-датчиков создаются дополнительные трудности в виде необходимости в предобработке видеоданных, полученных с таких камер, с целью устранения искажений видеоряда, что создает дополнительные требования для алгоритма, а именно возможность эти искажения устранять или корректно работать с учетом возможных искажений. Наиболее распространенные типы искажений, с которыми можно столкнуться при тестировании методов vSLAM на коллекции видео, полученных с таких камер являются Rolling shutter (искажения, возникающие из-за последовательного считывания данных с датчиков в матрице видеокамеры), эффект “рыбий глаз” (возникающий из-за дефектов в производстве объективов), дисторсия и шум (возникает из-за недостаточной освещенности снимаемой местности).

При использовании камер с ПЗС-датчиками, как правило полученный видеоряд превосходит по качеству аналогичный, полученный с камеры оснащенной КМОП-датчиком, что частично или полностью убирает необходимость в предобработке видеоряда перед тестированием методов vSLAM. Также, при использовании видео с количеством кадров более 30 в секунду, появляется возможность строить более детализированные карты за счет большего набора данных, но в то же время это создает дополнительную вычислительную нагрузку на программную реализацию алгоритма и, следовательно, на вычислитель выполняющий алгоритм.

Стереопара позволяет повысить точность работы алгоритмов за счет того, что камеры располагаются на известном расстоянии друг от друга и съемка каждого кадра происходит в один и тот же момент времени. Это позволяет повысить точность карты глубины для каждого кадра в видеоряде (а точнее, пары кадров), полученной с использованием методов vSLAM. С другой стороны, с учетом возросшего количества информации для обработки, создается дополнительная нагрузка на вычислитель. Также, в контексте робототехники,

стоит учитывать возможность сжатия видеопотока для последующей передачи на наземную станцию с целью обработки видеоряда уже на ней, если ресурсов вычислителя, установленного на робототехническое устройство, не хватает.

RGB-D камеры позволяют получать видеоряд с уже построенной картой глубины для каждого кадра. Подобная возможность возникает за счет использования в таких камерах помимо КМОП/ПЗС-датчиков еще и ИК-датчика. Коллекции данных, полученных с помощью RGB-D камер представляют собой видеоряд, каждому кадру которого соответствует расстояние от камеры до места, соответствующему каждому пикселю изображения. Это позволяет снять нагрузку по вычислениям карты глубины с программной реализации алгоритма vSLAM. Также появляется возможность строить высокодетализированные карты пригодные не только для навигации по видеопотоку, но и для 3D-реконструкции местности. Однако, подобные камеры имеют большие габариты, вес и энергопотребление, что делает их установку на робототехнические устройства малого типа – невозможной. Видео из коллекций, полученных с RGB-D камеры можно использовать для экспериментов с методами, рассчитанными на работу с данными, полученными с единственной камеры.

Для каждого из типа камер существуют разные коллекции данных, на которых можно осуществлять эксперименты с vSLAM. Коллекции также отличаются по содержанию видео. В открытом доступе имеются наборы видео различных типов местности (например, коридорного, уличного и т.д.), разной освещенности и погодных условиях (если речь идет об открытой местности). Все эти особенности коллекций позволяют протестировать методы vSLAM в разных условиях и оценить эффективность работы алгоритма в зависимости от типа местности.

Также, для оценки точности построенной траектории в некоторых коллекциях присутствует информация о реальной траектории, проделанной камерой во время съемки, а также позиции камеры в каждый момент времени. Такую информацию удастся получить за счет устройств захвата движения (motion

capture), которые используются параллельно со съемкой видео для коллекции. Такой подход имеет свои недостатки: полученная траектория все равно отличается от реальной, но этой разницей можно пренебречь в силу того, что среднее отклонение мало. Использование технологии motion capture позволяет строить траектории больших местностей и подходит для случаев, когда ручные замеры не могут быть произведены в силу того, что проделанная траектория во время съемок слишком велика.

Также, реальная траектория может быть получена с помощью GPS/ГЛОНАСС датчиков. С их помощью, при необходимости, можно замерить траекторию в несколько километров и более, что подходит для тестирования методов vSLAM в контексте управления наземными беспилотными транспортными средствами. Однако отклонение полученной траектории от реальной обуславливается точностью GPS/ГЛОНАСС датчиков, а также не позволяет отследить точное положение камеры в каждый момент времени, что может осложнить экспериментальную оценку эффективности работы алгоритмов vSLAM.

В качестве наиболее перспективных открытых коллекций данных для экспериментального исследования методов одновременного картирования и локализации по видеопотоку были выбраны 4 коллекции: RGB-D SLAM Dataset and Benchmark, LSD-SLAM Dataset, ICL-NUIM Dataset и Malaga Dataset 2009. Все коллекции имеются в свободном доступе и различаются как по формату видео, так и по его содержанию.

Коллекция **RGB-D SLAM Dataset and Benchmark** разработана Computer Vision Group Мюнхенского технического университета в рамках экспериментов с методом vSLAM Direct RGB-D SLAM. Коллекция представляет из себя набор видеозаписей, сделанных в помещении с помощью RGB-D камеры и состоит из 89-ти видеозаписей длиной от 25-ти секунд до 156-ти. Длина пройденной траектории варьируется от 1.5 метров до 20-ти. Видеозаписи сделаны с помощью камеры Microsoft Kinect.

В коллекции содержатся карты глубины для каждого видео, построенная с помощью Microsoft Kinect, что позволяет провести сравнение полученной методом vSLAM карты глубины для единственной камеры с имеющийся в коллекции. Максимальная длина видеозаписей не превышает 3х минут, а траектория – 20 метров, что не позволит проверить эффективность работы алгоритма на больших дистанциях. Коллекция содержит данных калибровки камеры таких как фокальное расстояние, координаты оптического центра и параметры дисторсии, что частично помогает устранить искажения на изображениях видеоряда.

Разработчики также предоставляют файлы в формате ROS Bag, что позволяет производить эксперимент в среде Robot Operating System (ROS), которая представляет собой фреймворк для работы с робототехническими устройствами и моделирования их поведения.

Наличие замкнутых траекторий позволяет протестировать эффективность алгоритмов корректировки накапливающихся ошибок траектории в методах vSLAM (loop-closure algorithms), а также проверить стабильность работы алгоритмов при многократных замыканиях. Присутствие движущихся объектов в видеоряде позволяет оценить влияние изменения окружающей среды на точность работы метода.

Коллекция предназначена на проведения экспериментов с алгоритмами vSLAM в помещениях комнатного типа при нормальном освещении.

LSD-SLAM Dataset был разработан Computer Vision Group Мюнхенского технического университета для тестирования метода LSD-SLAM как в помещениях, так и в условиях открытой местности. В данной коллекции присутствуют видеозаписи, произведенные при прохождении камерой больших траекторий с многократными замыканиями, что создает дополнительную нагрузку при тестировании алгоритмов картирования и локализации по видеопотоку в виде задачи по распознаванию уже пройденной траектории. Видео были записаны на

промышленную камеру без использования дополнительных датчиков.

Данная коллекция потенциально применима для тестирования методов vSLAM в условиях помещений и на открытой местности. Разработчики не использовали дополнительных методов для измерения пройденной траектории, кроме метода LSD-SLAM, что позволяет сравнивать результаты работы тестируемого алгоритма только с результатами работы метода LSD-SLAM.

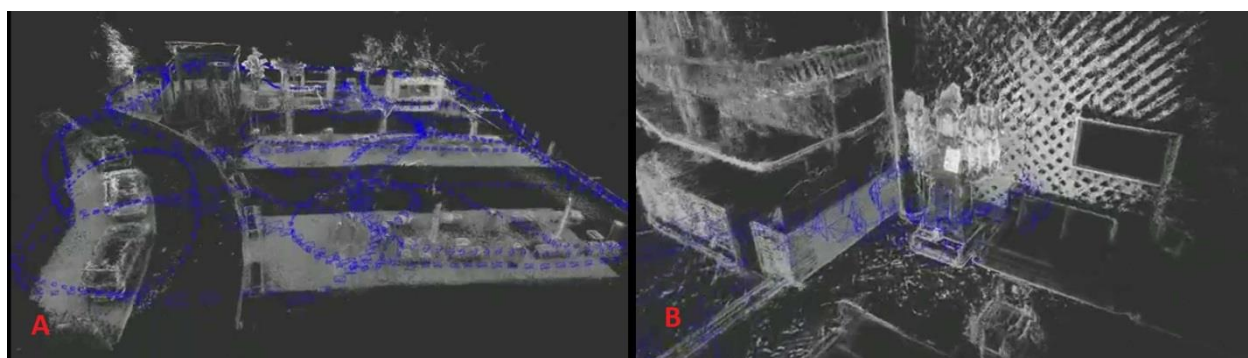


Рис. 9. Примеры карт и траекторий, построенных с помощью метода LSD-SLAM.

A- Foodcoutr Sequence, B- ECCV Sequence.

В следствии того, что траектория не была измерена дополнительными датчиками, то информация о пройденном камерой расстоянии – неизвестно. Сравнение возможно только в относительных величинах.

Разработчики LSD-SLAM Dataset предоставляют карту, полученную в результате работы метода LSD-SLAM в виде облака точек (point-cloud), что позволяет сравнивать точность и уровень детализации полученной карты тестируемого алгоритма.

Коллекция состоит из четырех видеозаписей: Desk Sequence, ECCV Sequence, Foodcourt Sequence и Machine Sequence, 2 из которых сделаны в маленьком жилом помещении при комнатном освещении, а 2 на улице при дневном освещении. На первых 2х видео камера находится в руке оператора, который совершает резкие движения при съемке, что делает траекторию, пройденную камерой – хаотичной,

что в свою очередь вносит дополнительную нагрузку на тестируемый алгоритм в виде непредсказуемости дальнейшего движения.

Разработчик предоставляет доступ к данным калибровки камеры, что позволяет построить карту глубины изображений видеоряда для последующего сравнения с картой глубины, полученной методом LSD-SLAM. Также в коллекции имеются файлы видео и облака точек карты в формате ROS Bag, что позволяет производить экспериментальное сравнение в среде ROS. Поскольку видео записано на промышленную камеру, то на нем отсутствуют распространенные для камер с КМОП-матрицей искажения, а, следовательно, снижаются требования к предобработке изображений с целью устранения шумов.

Коллекция **ICL-NUIM Dataset** была создана разработчиками из Imperial College для тестирования алгоритмов одновременного картирования и локализации по видеопотоку и 3D реконструкции местности. Видеозаписи были сделаны с помощью RGB-D камеры. В рамках данной работы, также были построены 3D модели местностей, в которых были произведены съемки с помощью ИК-датчиков.

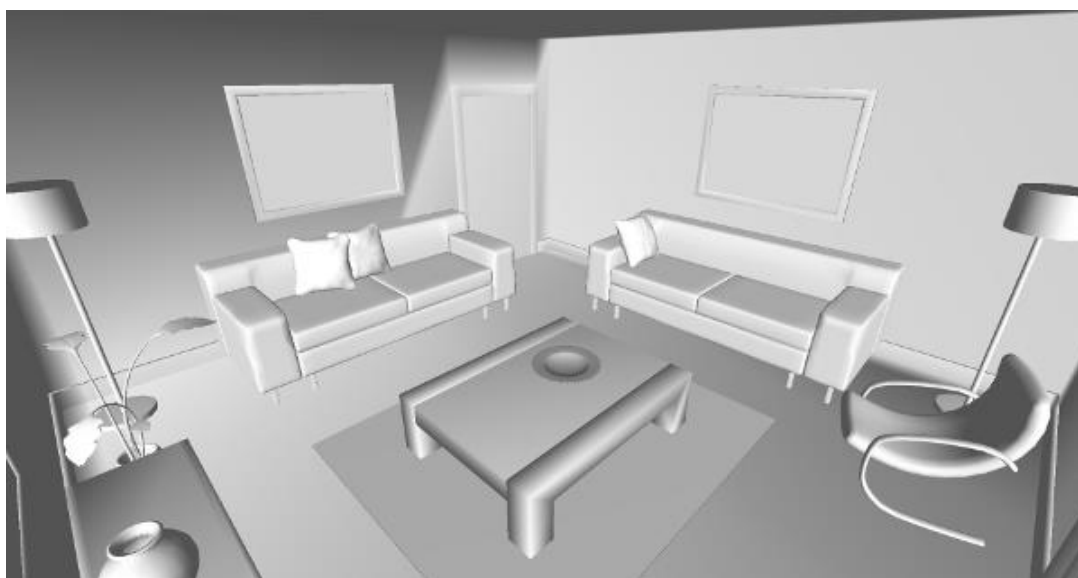


Рис. 10. 3D модель местности из набора видео Living Room Dataset

Измерение пройденной траектории, а также позиции камеры в каждый момент времени производилось в ручном режиме. Разработчики также предоставили данные в форматах TUM-RGB-D Compatible PNGs, что дает возможность присоединения данной коллекции к RGB-D SLAM Dataset и LSD-SLAM Dataset без дополнительных преобразований форматов.

Данная коллекция состоит из восьми видеозаписей, 4 из которых сделаны в жилом помещении и 4 в помещении офисного типа. На видео присутствуют искажения, характерные камерам с КМОП-матрицей. ICL-NUIM Dataset подойдет для тестирования алгоритмов одновременного картирования и локализации по видеопотоку в условиях небольших, замкнутых пространств. В коллекции отсутствуют видеозаписи с замыкающимися траектории, что делает невозможным эксперимент с корректировкой алгоритмом траектории при замыкании или следовании по уже пройденной траектории.

Также разработчиками была создана открытая программа-визуализатор SurfReg tool для отображения 3D моделей отснятых помещений и пройденных траекторий.

Malaga Dataset 2009 был разработан специально для тестирования методов vSLAM. Коллекция содержит в себе видеозаписи, сделанные с помощью промышленной стереопары, карту глубины для каждой пары изображений видеоряда, а также пройденную траекторию и карту местности, построенную с помощью GPS датчиков, лазерных сканеров и ИК датчиков. Разработчиками была создана специальная экспериментальная установка для точного измерения траектории, которая представляет собой электромобиль, оснащенный всеми вышеперечисленными датчиками.



Рис. 11. Экспериментальная установка

Коллекция Malaga Dataset 2009 подойдет для тестирования алгоритмов одновременного картирования и локализации в контексте навигации автономных наземных транспортных средств. При составлении коллекции было уделено особое внимание точности построения пройденной траектории в точности до сантиметра.

Также, разработчиками была реализована поддержка открытой библиотеки Mobile Robot Programming Toolkit, которая дает возможность визуализации и 3D моделирования поведения робототехнических устройств и методов vSLAM.

Для сравнения открытых коллекций, потенциально применимых для тестирования методов одновременного картирования и локализации по видеопотоку, полученному с камеры, необходимо выбрать часть критериев, которые позволят понять особенности каждой из коллекций и исходя из критериев можно судить о применимости конкретной коллекции для экспериментального исследования методов vSLAM. Также, выбранные критерии позволят судить о том, какая из коллекций лучше подойдет под моделируемые условия окружающей среды. Среди таких критериев можно выделить тип камеры и датчика, отсутствие/присутствие искажений, шумов и движущихся объектов, наличие карты глубины и способ моделирования реальной (ground-truth) окружающей среды и пройденной траектории.

Таблица 2. Параметры коллекций RGB-D SLAM Dataset and Benchmark, LSD-SLAM Dataset, ICL-NUIM Dataset и Malaga Dataset 2009

	RGB-D SLAM Dataset and Benchmark	LSD-SLAM Dataset	ICL-NUIM Dataset	Malaga Dataset 2009
Тип камеры	RGB-D	Промышленная	RGB-D	Стереокамера
Разрешение сенсора	640x480	640x480	640x480	640x480
Формат видео	avi	png	png	Набор пар JPEG изображений с отметкой времени
Тип датчика	КМОП/ИК	ПЗС	КМОП/ИК	ПЗС
Количество кадров в секунду	30	50	30	30
Способ моделировани я реальной пройденной траектории	Motion capture	Метод LSD-SLAM	3D моделировани е	Реконструкци я с помощью GPS- датчиков, лазерных сканеров и ИК-датчиков
Количество видеозаписей	89	4	8	6

Минимальная длина видео (мин)	0:25	0:55	0:30	1:24
Максимальная длина видео (мин)	2:26	12	0:51	3:11
Минимальная пройденная траектория (м)	1.5	-	1.7	89
Максимальная пройденная траектория (м)	20	-	5	342
Наличие замкнутых траекторий	+	+	-	+
Наличие карты глубины	+	+	+	+
Тип местности	Помещение	Помещение/Улица	Помещение	Дорога
Количество видео с присутствием движущихся объектов	9	0	0	0

Наличие данных калибровки камеры	+	+	+	+
Тип искажений	Rolling shutter, рыбий глаз, смазанность изображения при движении	-	Rolling shutter, рыбий глаз, смазанность изображения при движении	-
Возможность моделирования в среде ROS	+	+	+	-
Тип освещения	Комнатное	Комнатное/Дневное	Комнатное	Дневное

Как видно из таблицы 9, для тестирования методов vSLAM, использующих RGB-D камеру, необходимо использовать коллекции RGB-D SLAM Dataset and Benchmark и ICL-NUIM Dataset. Для алгоритмов, использующих стереокамеру, возможно использование Malaga Dataset 2009, содержащую в себе изображения как с левой, так и с правой камер. ICL-NUIM Dataset и Malaga Dataset 2009 хорошо подходят для тестирования аккуратности построенной методами vSLAM карт и траекторий, т.к. они содержат в себе точные реконструкции отснятых местностей и пройденных траекторий. На коллекциях LSD-SLAM Dataset, RGB-D SLAM Dataset and benchmark и ICL-NUIM Dataset возможно производить эксперименты в условиях помещений. Для уличных или дорожных условий лучше использовать наборы Malaga Dataset 2009 и часть видео из набора LSD-SLAM Dataset. Все

вышеописанные коллекции имеют набор данных для калибровки камеры, что позволяет проводить эксперименты с методами vSLAM, работающими с зашумленными и искаженными данными.

Несмотря на разнообразие открытых коллекций, среди существенных недостатков можно выделить невозможность произвольного перемещения робота по среде, поскольку данные уже записаны и статичны, а также вопрос связанный с определением, являются ли данные о карте и траектории робота истинными, поскольку датчики внутреннего и внешнего позиционирования имеют погрешность.

1.4 Выводы по главе

Использование единственной камеры для картирования и локализации накладывает свои ограничения. Из-за отсутствия привязки к каким-либо реальным объектам, в отличие от RGB-D SLAM, где карта глубин поступает в качестве входных данных или Stereo SLAM, где используется стереопара (расстояние между камерами – известно заранее), методы vSLAM с единственной камерой восстанавливают траекторию и строят карту в относительном масштабе. В задачах автономной навигации робототехнических устройств возникает ситуация, когда с помощью таких методов невозможно определить ключевые параметры в реальном масштабе, а именно текущую скорость, размеры препятствий и т.д. Т.е. становится затруднительно оперировать в построенной карте, поскольку неизвестно как размеры робота относятся к размерам карты. Следующей проблемой является низкая детализация построенных карт. Непрямые методы используют в качестве базовых элементов особенности изображений для построения карты неизвестной местности. Несмотря на то, что такие особенности обладают рядом преимуществ (однозначная идентификация, робастность к искажениям на изображении и т.д.), их на изображении выделяется мало относительно количества элементов изображения. В силу своей природы особенности изображений – это как правило

области (точки, углы, линии и т.д.) с наибольшим локальным градиентом, а это значит, что на слаботекстурированных областях особенности изображений не могут быть выделены и нанесены на карту. Это приводит к тому, что препятствие может быть не отражено на карте и робот при навигации с этим препятствием столкнется. В прямых методах наоборот, препятствие может быть нанесено туда, где его в реальности нет.

Поскольку задача mvSLAM связана с обработкой видеопотока, а специфика применения в робототехнике связана с нехваткой вычислительных мощностей, одной из ключевых проблем является малое быстродействие алгоритмов обработки видеопотока. Даже несмотря на быстроту работы непрямых методов, их использование в реальных задачах может быть осложнено. Одной из причин падения быстродействия является работа алгоритма поиска замыкания траектории. Поскольку в каждый момент времени текущее изображение сравнивается со всеми предыдущими, то вычислительная сложность с каждым новым изображением увеличивается. Последней проблемой является сложность оценки картирования методов, решающих задачу vSLAM, поскольку невозможно однозначно определить соответствие между двумя наборами элементов карт: истинного и построенного.

Основываясь на вышеизложенном, можно сделать вывод о том, что у современных методов, решающих задачу одновременного картирования и локализации по видеопотоку существуют проблемы, которые затрудняют их использование в задачах автономной робототехники. К основным проблемам можно отнести малую детализацию карт, относительный масштаб при построении карты и траектории (для непрямых методов), и быстродействие (для прямых методов). Все описанные проблемы нуждаются в полном или частичном устранении путем улучшения отдельных алгоритмов внутри методов, решающих задачу mvSLAM, и путем использования новых подходов к построению карт. Именно эти проблемы определяют задачи диссертационной работы.

Задачи диссертационной работы

На основе проведенного в Главе 1 анализа, были сформулированы следующие задачи диссертационной работы:

1. Разработать эвристический алгоритм обнаружения замыкания траектории для повышения точности и вычислительной эффективности методов решения задачи картирования и локализации по видеопотоку.
2. Разработать модель искусственной нейронной сети, позволяющей решать задачу восстановления глубины изображений в режиме реального времени с приемлемым качеством для последующего использования в задаче одновременного картирования и локализации по видеопотоку единственной камеры.
3. Построить модель оценки качества карт (трехмерных облаков точек), получаемых с помощью методов решения задачи mvSLAM, с учетом особенностей задачи одновременного картирования и локализации.
4. Провести экспериментальные исследования разработанных алгоритмов и моделей на открытых коллекциях данных, в симуляционных средах и на реальных робототехнических системах.

Глава 2. Разработка оригинального алгоритма поиска замыканий траектории

В процессе картирования и локализации, в силу особенностей входных данных (зашумленные и некачественные изображения), накапливаются ошибки вычислений позиции и карты местности, из-за чего падает качество vSLAM. Также, в реальных условиях при длительной работе накапливается ошибка, которая приводит к потере локализации, что ведет к невозможности дальнейшей навигации робототехнической системы.

Один из способов скорректировать описанные особенности и, таким образом, повысить качество работы методов, решающих задачу mvSLAM, - использовать алгоритмы обнаружения замыканий траекторий (см. Рис. 12). Т.е. необходимо иметь возможность определять изображения, которые соответствуют местам, в которых система уже была до текущего момента и по полученным данным производить коррекцию карты и траектории.

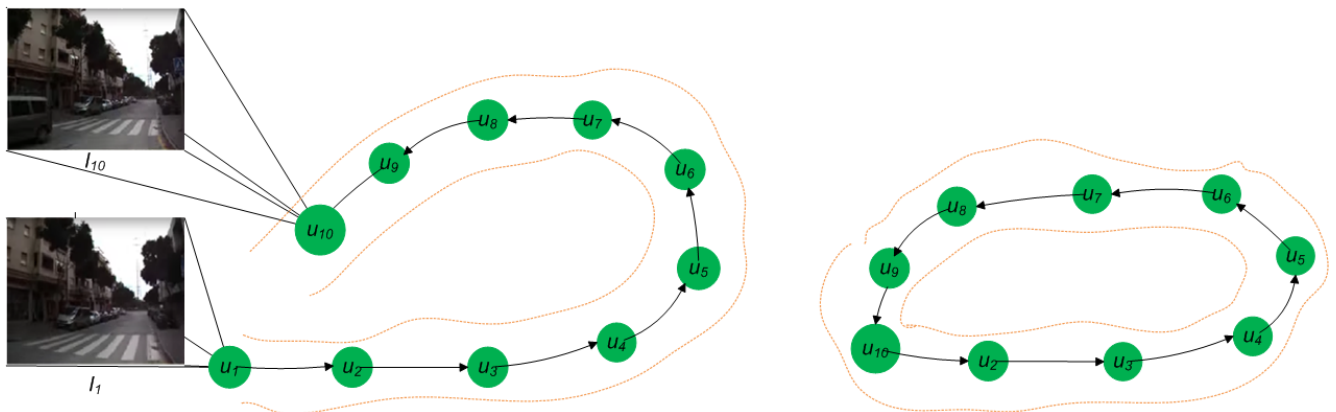


Рис. 12. Схематическая иллюстрация проблемы обнаружения замыканий на траектории (слева) и глобальной оптимизации (справа) с целью корректировки накопившийся ошибки. (Черная кривая – траектория, зеленые точки – позиции камеры, оранжевые точки – карта).

Все алгоритмы замыкания траекторий состоят из двух этапов: 1) обнаружение замыкания [2], 2) глобальная оптимизация [74]. Задачей процедуры поиска замыканий является обнаружение факта присутствия наблюдаемого изображения (или части сцены на изображении) на уже построенной карте. В результате определяются места, в которых робототехническая система уже была в процессе картирования и локализации. После обнаружения таких мест выполняется глобальная оптимизация трехмерной карты и полученной траектории с целью устранения накопившейся ошибки. В случае mvSLAM, задача сводится к распознаванию изображений (текущее изображение сравнивается с предыдущими) [41].

Упрощенный пример работы алгоритма замыкания траекторий изображен на (Рис. 13). Зелеными точками обозначены позиции $(u_1, u_2, \dots, u_{10})$, которые были получены в результате локализации. Оранжевые точки – карта. Черная кривая – траектория движения системы. В рассматриваемом сценарии расстояние между u_1 и u_{10} в действительности минимально. Однако, в силу специфики работы алгоритмов, решающих задачу mvSLAM, траектория и карта подвержены воздействию накапливающейся в процессе работы ошибки. Чтобы устранить эту ошибку, для каждой позиции $u_t, t = [1; 10]$ рассматривается изображение I_t , которое было получено в процессе работы методов, решающих задачу mvSLAM. На каждом шаге происходит сопоставление текущего изображения и его фрагментов с предыдущими.

Задачу сопоставления изображений можно сформулировать следующим образом: пусть даны изображения $I_1 = \{a_{ij}\}$ и $I_2 = \{b_{ij}\}, a_{ij}, b_{ij} \in \mathbb{N}, i, j \in \mathbb{N}, i \in [1; M], j \in [1; N]$, и функция выделения особенностей $F_{feature}: I \rightarrow \{\mathbb{R}^2 \times \mathbb{R}^n\}$. $F_{feature}(I) = \{(C, D)\}$, где $C = (i, j)$ – координаты особенности изображения (особой точки), $D = (d_1 \dots d_n)$ – элементы идентификатора (дескриптора) особой точки. Изображения I_1 и I_2 считаются сопоставленными, когда $|G(F_{feature}(I_1), F_{feature}(I_2))| > H, H \in \mathbb{N}$ – пороговое значение, G – функция сопоставления. Функция сопоставления ищет одинаковые или

близкие вектора из двух множеств векторов. Одним из примеров функции служит функция сопоставления методом ближайшего соседа.

Если найдено хотя бы одно совпадение текущего изображения с одним из предыдущих, то считается, что соответствующие сопоставленным изображениям позиции находятся рядом или совпадают и на этапе глобальной оптимизации происходит корректировка траектории и построенной карты с целью устранения накопившийся ошибки (рис 10).

Глобальная оптимизация выполняется после обнаружения замыкания с учетом соответствующих изображениям позиций. Этот шаг компенсирует накопленную ошибку во время выполнения как для карты, так и для траектории в фоновом режиме.

Алгоритмы поиска замыканий на траектории является триггером для запуска алгоритма глобальной оптимизации. Алгоритм поиска замыканий на траектории запускается для каждого нового входящего изображения. Основным его недостатком является необходимость в сопоставлении текущего изображения со всеми предыдущими, что приводит к увеличению вычислительной сложности по мере работы алгоритма.

Алгоритм глобальной оптимизации запускается после нахождения замыкания на траектории. Оптимизация происходит как для траектории, так и для трехмерной карты, что ведет к изначальной вычислительной сложности алгоритма, которая не позволяет запускать его на каждом шаге (в отличия от поиска замыканий на траектории), иначе возможно нарушение работы метода, решающего задачу vSLAM в реальном времени, особенно на бортовых вычислителях.

В рамках диссертационной работы предложено улучшения алгоритма поиска замыканий на траектории, поскольку суммарно он работает дольше, чем алгоритм глобальной оптимизации, а значит и его улучшение принесет больший прирост в быстродействии.

2.1 Алгоритм поиска замыканий на траектории

Основным недостатком алгоритма поиска замыканий на траектории является необходимость в сравнении текущего изображения со всеми предыдущими. По мере работы алгоритма поступают все новые изображения, которые (как правило в виде набора из особенностей) хранятся в памяти. Таким образом, после длительной работы, сравнение вновь поступившего на вход алгоритму изображения становится все более ресурсоемкой задачей.

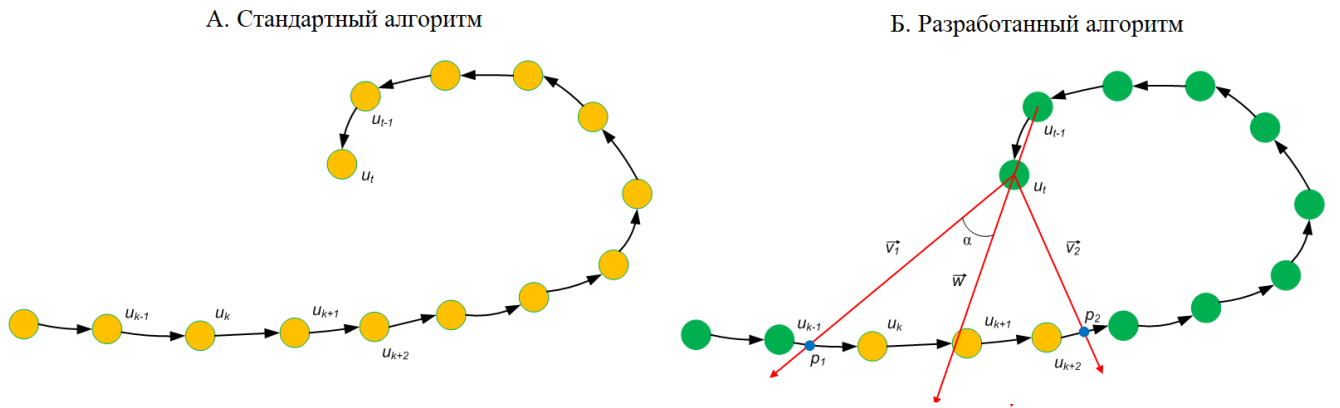


Рис. 13. Иллюстрация области поиска замыканий траектории

Основным улучшением, предложенным в диссертационной работе – использование геометрических ограничений на движения робота с целью минимизации области поиска замыканий на траектории (поскольку каждому изображению соответствует позиция робота на карте). Это позволяет идентифицировать участок на всей траектории, который с высокой вероятностью содержит точку замыкания траектории (то есть место, где робот уже находился).

В работе рассматривается двумерный случай, который без потери общности может быть расширен на трехмерную постановку задачи. Пусть даны позиции U , полученные после картирования и локализации для каждого момента времени t и последовательность изображений $I = \{I_1, I_2, \dots, I_T\}$, $I_t = \{a_{ij}\}$, $i \in [1; M]$, $j \in [1; N]$, $t \in [1; T]$, $i, j, t \in \mathbb{N}$. Для каждого изображения определен соответствующий набор особенностей $F_{feature}(I_t) = \{(C, D)\}_t$, который используется для

сопоставления. Для каждого u_t , выполняется построение ограничивающего окна с помощью двух векторов $\vec{v}_1 = sR(\alpha)\vec{w}$, $\vec{v}_2 = sR(-\alpha)\vec{w}$, где $\vec{w} = \overline{u_{t-1}, u_t}$, $s > 0$ – коэффициент масштаба (длина вектора, подбирается вручную), $R(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$ – матрица поворота на угол α .

После построения ограничивающего окна происходит проверка, имеются ли пересечения между любыми из векторов $\vec{u} = \overline{u_{t-1}, u_t}$ с векторами \vec{v}_1 и \vec{v}_2 . Если пересечение есть, то обозначим его как p_1 и p_2 соответственно. Ближайшие позиции к точкам пересечения p_1 и p_2 и все позиции и соответствующие им изображения будут областями поиска для алгоритма замыкания траекторий.

Иными словами, строится вектор направления, проходящий через предыдущую и текущую позиции. Затем находится перпендикуляр к этой линии. Если перпендикуляр пересекает построенную траекторию, то строится угол α между нормалью и линией проекции движения. Ближайшая позиция (с соответствующим изображением) к точке пересечения будет отправной точкой для алгоритма обнаружения траекторий, причем вся область обнаружения траекторий ограничена двумя точками - пересечением нормали и линии движения с траекторией. Если линия движения не имеет точки пересечения, то на вход алгоритму поступает следующее изображение. Иллюстрация предлагаемого метода показана на Рис. 13.

Было предложено ограничить область поиска угловым окном относительно проекции вектора движения камеры. Варьируя угол окна α можно увеличивать или уменьшать область поиска соответствий на изображениях. Например, при $\alpha = 2\pi$ получаем классический подход, где текущее изображение сравнивается со всеми предыдущими.

2.2 Сопоставление изображений

Сопоставление изображений влияет на общее время работы методов, решающих задачу mvSLAM. Как правило, принцип сопоставления изображений происходит в несколько этапов. На первом этапе для двух изображений выделяются их особенности (особые точки). Для каждой особой точки рассчитывается дескриптор, который является однозначным идентификатором ключевой точки. Т.е. с каждым изображением связан набор дескрипторов. Далее, набор дескрипторов текущего изображения сравнивается с набором дескрипторов одного из предыдущих изображений изображения и если количество совпадений больше заранее заданного порога, то изображения считаются сопоставленными.

Для быстрого и точного сопоставления изображений было экспериментально установлено, что сохранение определенного количества информативных ключевых точек (вместо всех ключевых точек) для каждого изображения позволяет сохранять точность сопоставления. Критерием информативности ключевой точки является сумма значений ее дескриптора. Информативной точкой будет точка, сумма значений дескриптора которой – максимальна. Поскольку значения дескриптора является характеристикой контрастности области вокруг точки на изображении, а точка с наибольшими значениями дескриптора в некоторой конкретной области наиболее контрастная, по сравнению с другими точками, и такая точка лучше идентифицирует конкретное изображение, нежели точка, у которой значения дескриптора – минимальны.

Улучшение алгоритма состоит в том, что для каждого изображения отбираются K точек, у которых $\sum_{i=1}^L d_i \rightarrow \max$, по которым потом и происходит сопоставления.

2.3 Реализация алгоритма поиска замыканий на траектории в составе комплекса программной системы vSLAM

Поскольку разработанный алгоритм поиска замыканий на траектории является универсальным как для прямых методов, так и для непрямых, то он был реализован для методов ORB-SLAM и LSD-SLAM. В ORB-SLAM дополнительное выделение особенностей изображений не использовалось, поскольку алгоритм уже содержит данные по каждому изображению. Эти данные и были использованы в процессе работы алгоритма поиска замыканий на траектории. Для метода LSD-SLAM выделение ключевых точек происходит для каждого нового изображения дополнительно, поскольку в процессе работы метода LSD-SLAM выделение особенностей не используется.

В качестве основного идентификатора изображения для обнаружения замыканий был выбран детектор ORB [94] как один из самых быстрых, надежных и эффективных детекторов особых точек. Для каждого изображения извлекается как минимум K особенностей и сохраняем их дескрипторы, сумма элементов которых превышает порог, со связанными изображениями. Точки с большой суммой элементов имеют больший градиент интенсивности в этих точках, что обеспечивает более надежное сопоставление объектов. Это означает, что ключевые точки являются информативными и могут быть сохранены для дальнейшего сопоставления изображений.

Для глобальной оптимизации карт и траекторий, был использован один из самых популярных и эффективных наборов программных средств для оптимизации графов g^2o [35]. Это позволяет сохранять высокую точность при оптимизации карты и траектории по сравнению с другими современными методами, решающими задачу mvSLAM.

Псевдокод алгоритма обнаружения замыканий с вышеупомянутыми встроенными процедурами показан ниже.

1. **Входные данные:** текущее изображение I_t , позиции X , карта M , ключевые точки и их особенности $\{C, D\}_1, \dots, \{C, D\}_{t-1}$
2. $\{C, D\}_t = F_{feature}(I_t)$
3. $i = 0$
4. $(left, right) = window(x_t)$
5. **If** $left \neq 0$ **AND** $right \neq 0$
6. Отобразить K дескрипторов из текущего изображения I_t
7. **For** i **in** $[left; right]$
8. **If** $G(F_{feature}(I_t), \{C, D\}_i) > K$
9. $F_{global}(X, M)$
10. **End if**
11. **End for**
12. **End if**

На вход алгоритму поступает текущее изображение и особые точки с дескрипторами всех предыдущих изображений, текущая траектория и карта. На первом этапе (строка 2) происходит выделение особенностей текущего изображения и расчет их дескрипторов. Далее, происходит проверка геометрических ограничений (строка 4) и вычисление границ области поиска замыканий $left$ и $right$. Если границы были найдены, то для текущего изображения отбираются K информативных дескрипторов и эти дескрипторы сопоставляются с предыдущими информативными дескрипторами.

Если сопоставление найдено, то запускается алгоритм глобальной оптимизации (строка 9). Если нет – то алгоритм повторяется для следующего изображения.

2.4 Результаты полунатурного эксперимента

Все эксперименты проводились с использованием набора данных LSD-SLAM Dataset <http://vision.in.tum.de/research/vslam/lslam>, эталонного теста KITTI [32] и набор данных Malaga [5]. Видеофрагменты были разделены на последовательности (для каждого эксперимента последовательности различаются), чтобы сделать экспериментальные исследования более разнообразным.

Первый эксперимент был проведен для проверки минимального количества необходимых дескрипторов информативных точек (значение порога K) для успешного функционирования алгоритма обнаружения замыканий в части сопоставления изображений. В качестве тестовых данных использовались видеопоследовательности Sequence 13, 14, 15, Machine и Foodcourt коллекции LSD-SLAM Dataset.

Для эксперимента вручную были отобраны пары изображений с повторяющихся участков видеопоследовательностей Sequence 13, 14, 15, Machine и Foodcourt таким образом, чтобы визуально на обоих изображениях присутствовало более 50% перекрытия объектов из коллекции. Для каждого изображения выделялся набор информативных точек, а затем происходило сопоставление. Если сопоставление происходило успешно, то число информативных точек уменьшалось на 1 (откидывалась точка с минимальной суммой элементов дескриптора) для каждого изображения и сопоставление производилось снова до тех пор, пока не было достигнуто минимальное число информативных точек, при которых было возможно сопоставление.

Результаты успешных и неуспешных сопоставлений изображений в зависимости от количества информативных точек представлены в таблице 2.

Таблица 3. Таблица срабатываний метода поиска замыканий траекторий в зависимости от количества особых точек K .

Коллекция	Метод	Количество особенностей ORB										
		8	9	10	11	12	13	14	15	16	17	18
Sequence 13	ORB-SLAM2	-	-	-	-	+	+	+	+	+	+	+
	LSD-SLAM	-	-	-	-	-	-	+	+	+	+	+
Sequence 14	ORB-SLAM2	-	-	+	+	+	+	+	+	+	+	+
	LSD-SLAM	-	-	-	+	+	+	+	+	+	+	+
Sequence 15	ORB-SLAM2	-	+	+	+	+	+	+	+	+	+	+
	LSD-SLAM	-	+	+	+	+	+	+	+	+	+	+
Machine	ORB-SLAM2	-	-	-	-	-	-	+	+	+	+	+
	LSD-SLAM	-	-	-	-	-	-	-	+	+	+	+
Foodcourt	ORB-SLAM2	-	-	-	-	-	+	+	+	+	+	+
	LSD-SLAM	-	-	-	-	-	-	+	+	+	+	+

Экспериментальные данные показывают, что $K = 15$ является минимальным значением для обнаружения замыкания. Представленный результат также позволяет значительно сократить использование памяти, поскольку не нужно хранить сотни дескрипторов.

Для оценки критериев разработанного алгоритма поиска замыканий на траектории при вычисленном K было необходимо подобрать большое количество реальных траекторий из открытых коллекций данных. В связи с этим, изначальный

набор видеоданных был значительно расширен до 41 видеопоследовательности. Видеопоследовательности взяты из следующих коллекций: Malaga Dataset 2009, Malaga Dataset 2013, LSD-SLAM Dataset, KITTI Dataset. За счет того, что в коллекциях Malaga Dataset 2009 и 2013 присутствуют продолжительные траектории с множественными замыканиями, для увеличения экспериментальной выборки такие траектории были разбиты на более короткие. Таким образом, видеопоследовательность Parking 6L была разбита на 6 последовательностей, Parking 2L – на 2, Extract 05- на 2, Extract 06 – на 2, Extract 07- на 2, Extract 08 – на четыре, Extract 10 – на 6 и Extract 13 – на две. Видеопоследовательности из LSD-SLAM Dataset и KITTI Dataset на последовательности не разбивались. На каждой из последовательности присутствует как минимум одно замыкание.

Для тестирования производительности и точности разработанного метода был использован набор программных средств Robot Operating System (ROS) [75], который предоставляет инструменты для исследований роботизированных алгоритмов в целом и для тестирования mvSLAM на основе видеопотока в частности. Реализации ORB-SLAM2 и LSD-SLAM были использованы из официальных репозиториях разработчиков.

Минимальная длина траектории среди всех видеопоследовательностей – 15.3 м., максимальная – 13 069 м. Количество точек на карте варьируется от 3 582 до 84 001 869. Исследовались методы ORB-SLAM2 и LSD-SLAM и их модифицированные версии с замененным алгоритмом поиска замыканий на траектории на разработанный (ORB-SALM2 + LC и LSD-SLAM + LC соответственно). Среди критериев для оценки были относительная средняя ошибка перемещения $TE = \frac{1}{N} \frac{1}{\sum x_N} \sum_{i=1}^N |x_i - x'_{i-1}| * 100$ (в %), средняя ошибка поворота $RE = \frac{1}{N} \sum_{i=1}^N |r_i - r'_i|$ (в град./м.), средняя ошибка траектории $ATE = \frac{1}{N} \sum_{i=1}^N (x_i - x'_i)^2$ (в м.), относительная ошибка траектории $RPE = \sum_{i=2}^N ||r_i(x_i - x_{i-1}) - r'_i(x'_i - x'_{i-1})||$ (в м.) и среднее время поиска замыканий на траектории в сек. x, r – позиция и угол поворота, вычисленный методом, решающим задачу mvSLAM, x', r' – истинные позиция и угол поворота.

Подробные результаты экспериментов представлены на Рис. 14-Рис. 18

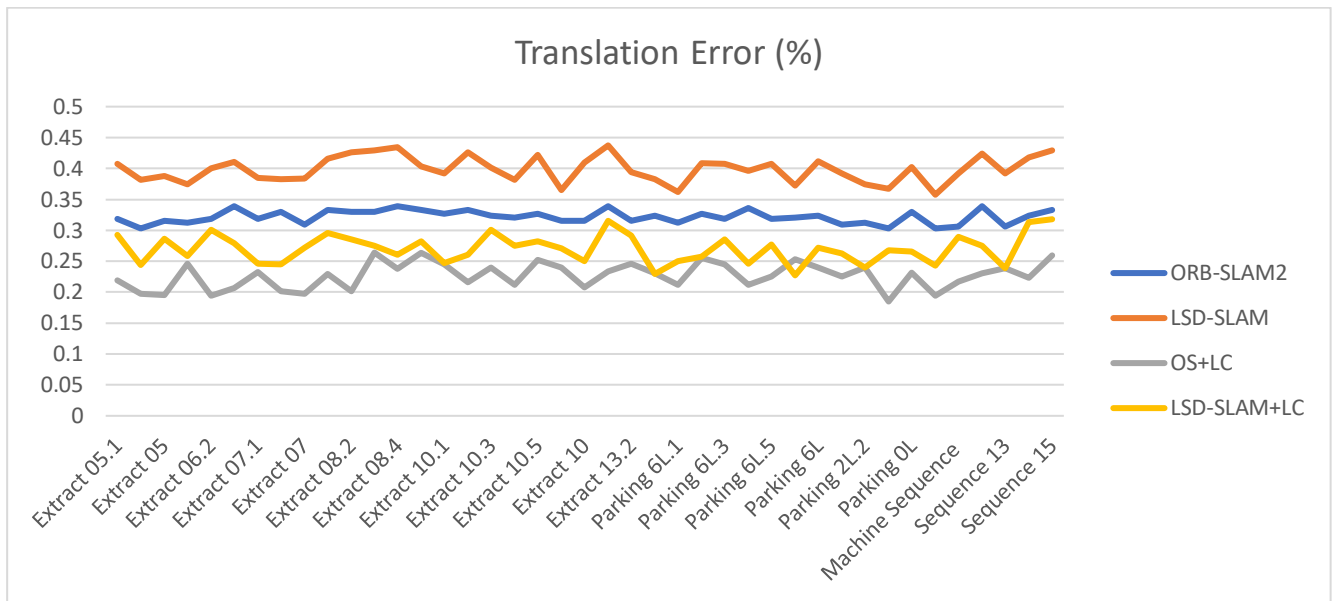


Рис. 14. График относительной ошибки траектории для всех коллекций в %

На Рис. 14 представлена ошибка траектории в процентах относительно общей длины траектории (ось Y). На оси X изображены использованные коллекции. Чем меньше значение – тем лучше. Из графика видно, что улучшенный алгоритм обнаружения замыкания траектории для методов LSD-SLAM и ORB-SLAM2 позволяет уменьшить ошибку для всех тестируемых последовательностей. В отдельных случаях получается добиться сравнимой ошибки траектории для метода ORB-SLAM2 по сравнению с LSD-SLAM. В оригинальной реализации метод ORB-SLAM2 этому параметру хуже, чем LSD-SLAM во всех случаях.

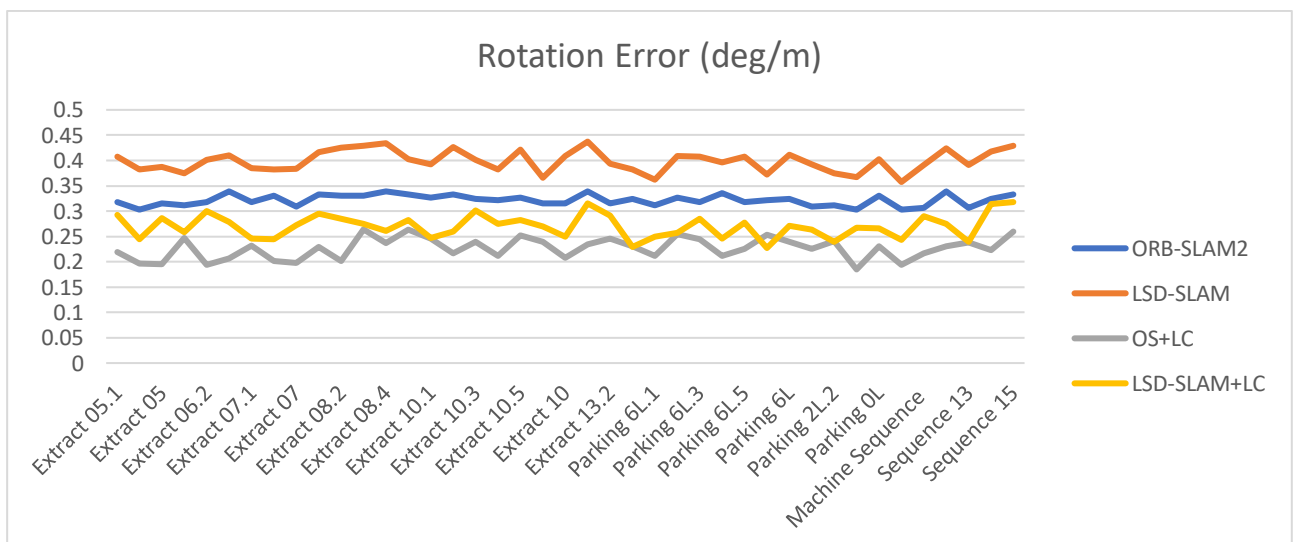


Рис. 15. График ошибки поворота позиции в град./м. для всех траекторий

На Рис. 15 представлены результаты ошибки поворота в градусах на метр (ось Y). Чем меньше значение – тем лучше. Результаты показали, что с помощью разработанного алгоритма удалось также улучшить точность алгоритмов LSD-SLAM и ORB-SLAM2 и в случае с ошибкой траектории удалось приблизить качество ORB-SLAM2 к LSD-SLAM, а в некоторых случаях добиться повышения качества.

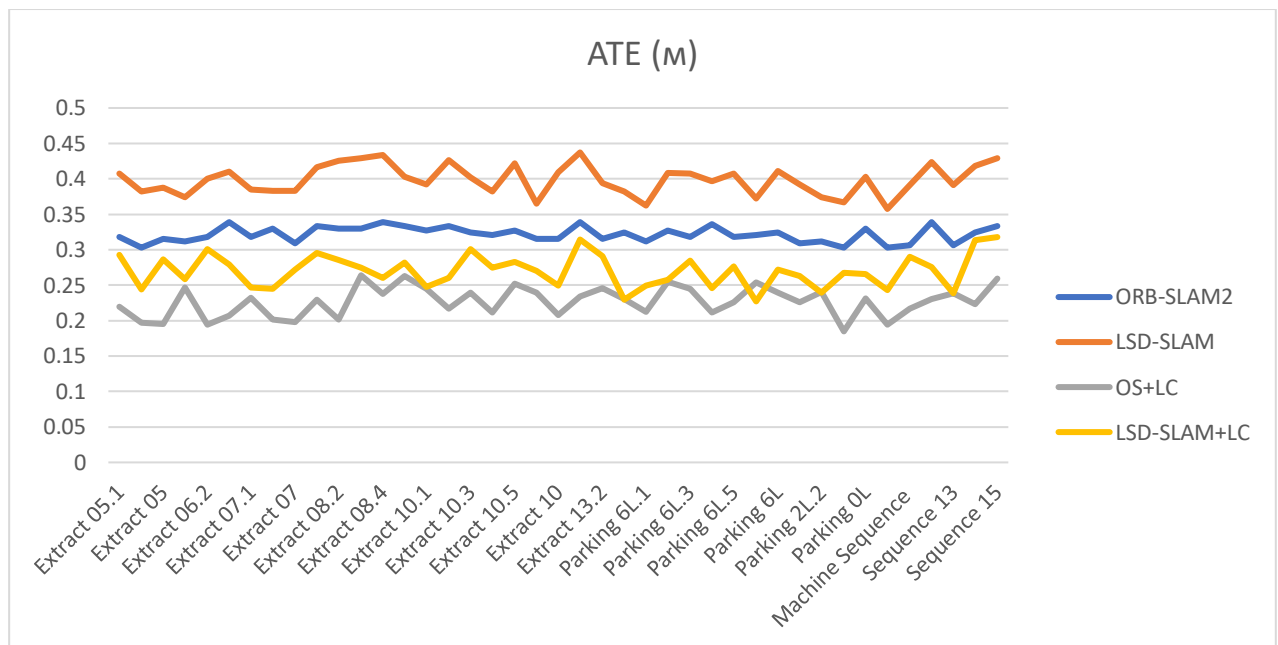


Рис. 16. Абсолютная ошибка траектории для всех коллекций в метрах.

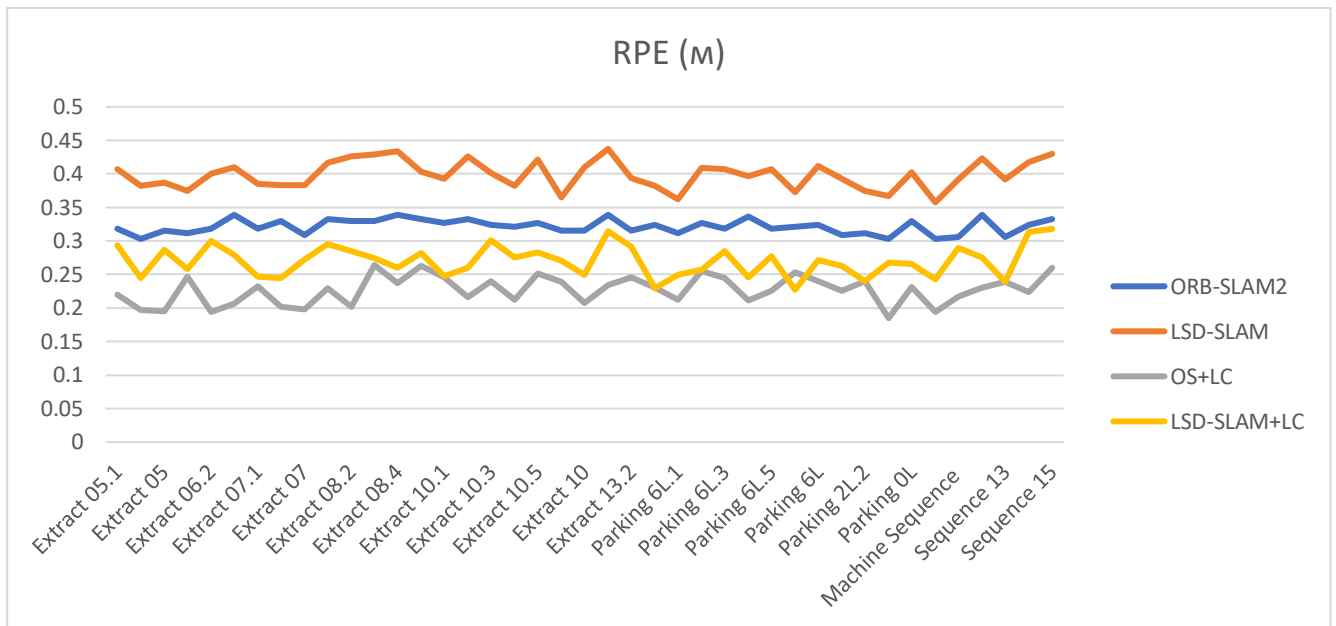


Рис. 17. График относительной ошибки траектории для всех коллекций в метрах.

На Рис. 16-Рис. 17 представлены абсолютная ошибка траектории (ось Y) в метрах и относительная ошибка траектории (ось Y) в метрах. Из графиков видно, что удалось также получить более качественную траекторию в абсолютных значениях, по сравнению с оригинальными методами. Однако добиться лучшего качества локализации метода ORB-SLAM2 по сравнению с LSD-SLAM удалось только в одном случае на видеопоследовательности Parking 6L.5.

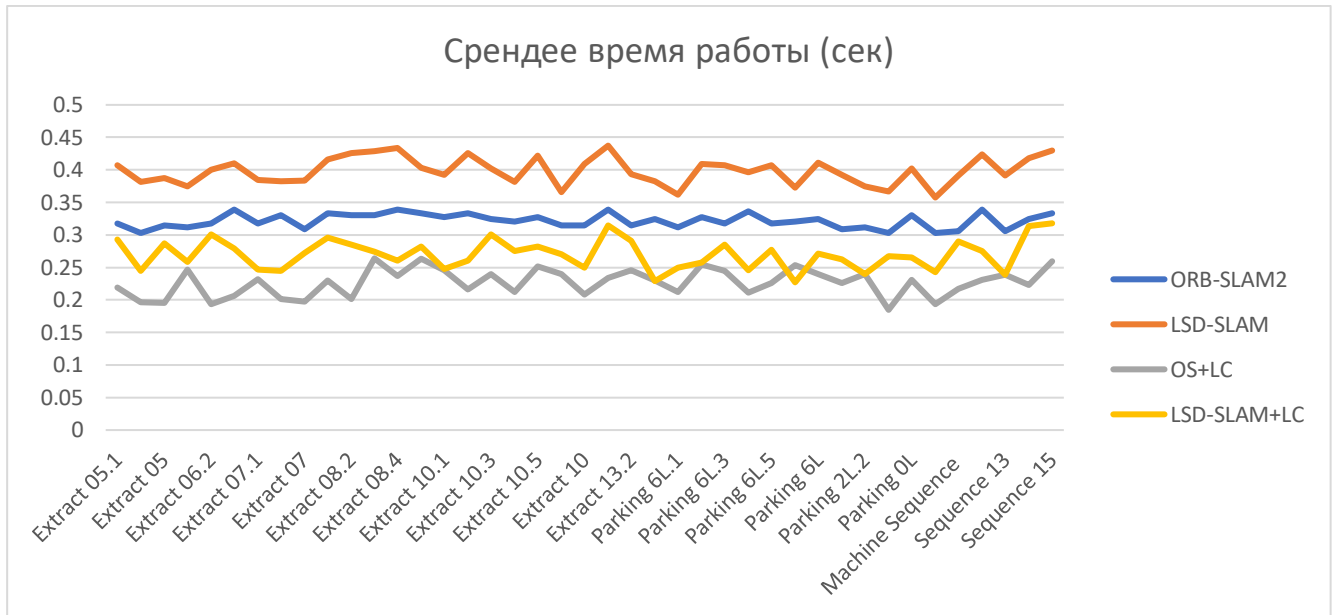


Рис. 18. Среднее время работы алгоритма поиска замыканий на траектории в секундах для всех коллекций.

На Рис. 18 представлено время работы алгоритмов в секундах (ось Y). Из графика видно, что удалось улучшить время обработки одного кадра для методов LSD-SLAM и ORB-SLAM2 по сравнению с оригинальными реализациями.

Таблица 4. Средние результаты тестирования разработанных алгоритмов по сравнению с оригинальными

	ORB-SLAM 2	LSD-SLAM	OS2+LC	LSD+LC
Среднее TE (%)	3.85	2.99	3.58	2.75
Среднее RE (deg/m)	0.10	0.09	0.09	0.08
Среднее ATE	0.85	0.75	0.77	0.70
Среднее RPE	0.88	0.85	0.81	0.81
Среднее время работы	0.32	0.40	0.23	0.27

В таблице 3 представлены средние данные по работе оригинальных алгоритмов (ORB-SLAM2, LSD-SLAM), а также их улучшенных версий с реализацией авторского алгоритма поиска замыканий на траектории (OS2+LC для ORB-SLAM2, LSD+LC для LSD-SLAM).

Таблица 5. Улучшение показателей относительно оригинальных методов в %

	OS2+LC	LSD+LC	Среднее
Среднее TE	7.49	8.96	8.23
Среднее RE	7.53	8.04	7.79
Среднее ATE	9.67	7.17	8.42
Среднее RPE	8.38	3.88	6.13
Среднее время работы	39.54	47.53	43.53

Относительные данные представлены в таблице 4. Основной задачей являлось улучшение времени работы без потери в качестве построения траектории и карты. Средний прирост производительности поиска замыканий траектории составляет 43.53%. Для ORB-SLAM2 прирост – 39.54%, для LSD-SLAM – 47.53%. Разница обуславливается тем, что для ORB-SLAM2 нет необходимости в дополнительном выделении особенностей на изображениях, поскольку эта процедура уже проделана и полученные ключевые точки с их дескрипторами могут быть еще раз использованы без дополнительных затрат. В LSD-SLAM необходимо дополнительно выделять особенности, поскольку эта процедура на этапе работы алгоритма нигде не используется.

Для того, чтобы проверить качество работы методов, решающих задачу mvSLAM, были получены метрики TE, RE, ATE и RPE для оценки качества траектории и CC для оценки качества картирования. Как оказалось, предложенные улучшения позволили не только сохранить качество построения траектории и карты оригинальных методов, но и улучшить их. Средняя относительная ошибка траектории была уменьшена на 7.49% для ORB-SLAM2 и на 8.96% для LSD-SLAM. Ошибка траектории уменьшилась на 8.23%. Средняя ошибка поворота уменьшилась на 7.79%, однако из-за того, что эта ошибка изначально была мала (0.1 град/м), то прирост качества по этой метрике – незначительный.

Средняя ошибка траектории была уменьшена на 8.42% по сравнению с оригинальными методами. Полученные данные позволяют судить о том, что помимо существенного прироста к производительности алгоритма поиска замыканий на траектории, удалось также добиться ощутимого прироста в точности

работы методов, решающих задачу mvSLAM, за счет более точного сопоставления изображений внутри ограничивающего окна, и следовательно, более качественных входных данных для алгоритма глобальной оптимизации.

2.5 Выводы по главе

В ходе исследования методов, решающих задачу одновременного картирования и локализации, было обнаружено, что одним из компонентов, существенно снижающий общую производительность, является алгоритм поиска замыканий траектории. Для улучшения быстродействия, а также увеличения качества картирования и локализации, был разработан эвристический алгоритм ограничения области поиска изображений на основе истории движения робототехнической системы (камеры). Предложен подход к ограничению количества сравнений особенностей изображений без потери качества сопоставлений, что привело к общему повышению производительности методов, решающих задачу mvSLAM.

Полученные улучшения были программно реализованы и интегрированы в известные алгоритмы ORB-SLAM2 и LSD-SLAM. Разработанные модификации были исследованы на открытых коллекциях данных, полученных с использованием реальных робототехнических систем, камер и датчиков глубины. Были исследованы основные метрики оценки качества локализации (отклонение траектории, поворота камеры, абсолютной позиции и т.д.), так и построения карты. Эксперимент показал, что разработанный алгоритм позволил улучшить качество построения траектории на 8,42% и производительность (скорость работы) на 43,53% по сравнению с оригинальными алгоритмами, использующимися в методах для картирования и локализации по видеопотоку.

Глава 3. Разработка вычислительно эффективных нейросетевых моделей для восстановления глубин изображений с последующим использованием в методах одновременного картирования и локализации

Одним из подходов к решению задачи одновременного картирования и локализации по видеопотоку (mvSLAM) с помощью монокулярной камеры является сведение ее к задаче картирования и локализации по данным RGB-D камеры (RGBD-SLAM) [80,83] с помощью нейросетевых моделей восстановления глубины изображений. Основанием является то, что RGB-D датчики, как правило, дорогие, габаритные, тяжелые и потребляют большое количество электроэнергии, что ограничивает их область применения, если речь идет о малых мобильных роботах. Ограничение привело к повышенному интересу к методам и алгоритмам для восстановления глубины с использованием единственной камеры, поскольку камерами, как правило, оснащаются все робототехнические системы из-за их легкости, компактности и энергоэффективности.

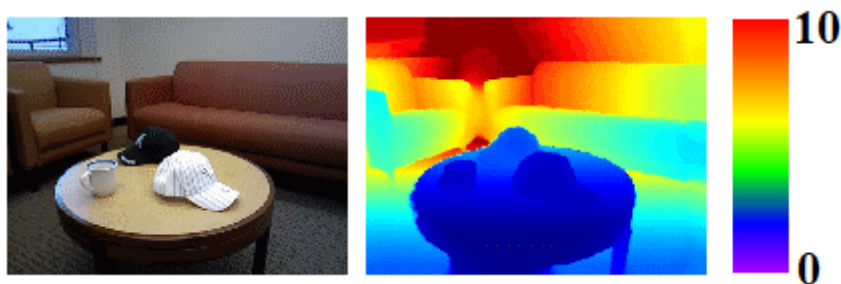


Рис. 19. Картинка (слева) и соответствующая ей карта глубины (справа).

Значения каждого пикселя карты глубины является расстоянием до
советующего объекта.

В настоящее время существуют методы машинного обучения и архитектуры сверточных нейронных сетей, которые позволяют восстанавливать глубину изображения с приемлемой для задач мобильной робототехники точностью [30,58]. Как правило, авторы этих методов сконцентрированы на точности восстановления,

в то время как скорость работы выходит за рамки исследований. Поэтому, такие методы не могут быть использованы в первозданном виде для создания полностью автономных робототехнических систем, которые не несут на борту мощных вычислителей, а имеют встраиваемый компьютер, пусть даже специально и предназначенный для задач обработки изображений, такой как NVidia Jetson TX2 [33,19]. Существующие современные методы преимущественно используют светочные нейронные сети для восстановления одометрии по двум последовательным кадрам [24,95], что не позволяет решить проблему детализированности карт, но работает в реальном времени на встраиваемых системах. С другой стороны, методы, использующие полносверточные нейронные сети для восстановления глубины с последующим картированием и локализацией либо не воспроизводимы в контексте исходного кода [55,103], либо не работают в реальном времени на встраиваемых системах [6,18,64].

Использование нейросетевых моделей на встраиваемых платформах. В последнее время обретают популярность встраиваемые системы, которые спроектированы для задач, решаемых нейросетевыми моделями. Одной из таких платформ является NVidia Jetson TX2, которая используется для ускорения обработки изображений на графическом процессоре в задачах семантической сегментации [72], распознавания объектов, классификация изображений и т.д. Размер и энергопотребление такой платформы позволяет использовать ее в качестве бортового вычислителя робототехнической системы. Доступность графического ускорителя стало решающим фактором при выборе NVidia Jetson TX2 в качестве целевой платформы, используемой в представленной работе для исследований полносверточных нейронных сетей (Fully convolutional neural networks – FCNN).

В настоящий момент, уже существует исследование на тему использование FCNN для восстановления глубины изображений на платформе NVidia Jetson TX2 в реальном времени [48]. Авторы используют облегченную архитектуру кодировщик-декодировщик, которая была обучена с помощью переноса знаний с

более глубокой архитектуры. Было достигнуто быстроедействие в 30 кадров в секунду с сравнимой с актуальными методами восстановления глубины точностью. Но исследования применимости в задаче mvSLAM не проводились (только устранение дрейфа масштаба карты), а результаты работы – не воспроизводимы (в контексте исходного кода или бинарных файлов). Исследование, представленное в диссертации нацелено на применение нейронных сетей для восстановления глубины изображений в реальном времени на встраиваемой системе. Главной задачей является применение в методах, решающих задачу mvSLAM, и воспроизводимость полученных результатов.

Возникает необходимость разработки метода восстановления глубины, основанного на моделях искусственных нейронных сетей, который: а) обеспечивает точность работы, достаточную для использования в методах одновременного картирования и локализации, б) способен функционировать в режиме реального времени на встраиваемой платформе, в) открыт сообществу, т.е. имеет открытый исходный код в виде узла для набора программных средств Robot Operating System (ROS).

3.1 Разработанные нейросетевые модели

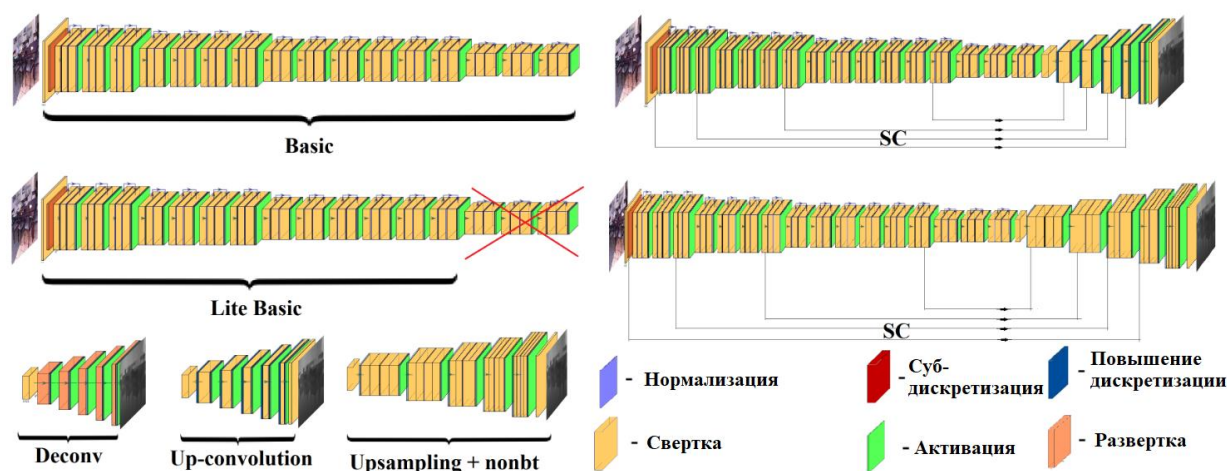


Рис. 20. Визуализация исследованных архитектур.

Были исследованы разные модификации архитектур полносверточных нейронных сетей для задачи восстановления глубины изображений. Для достижения режима реального времени реализован ряд алгоритмических улучшений, позволивший повысить быстродействие и сохранить точность работы алгоритмов.

Описание архитектур. Описание типовой архитектуры для восстановления карты глубины изображения дано в главе 1.2. В качестве основного кодировщика используется ResNet50 [99] (и его облегченная версия) и несколько разных модификаций декодировщика. Также, были предложены некоторые улучшения для увеличения быстродействия блоков кодировщика. Были получены 6 целевых архитектур для исследований.

Кодировщик. Кодировщик в искусственных нейронных сетях используется для получения кодированного представления изображения в виде набора особенностей путем серии математических операций, таких как свертка, субдискретизация, активация и др.

Основные операции кодировщика можно описать следующим образом. Пусть определен тензор размера $H \times W \times C$: $T = \{p_{h,w,c} \mid p_{h,w,c} \in \mathbb{R}, h \leq H, w \leq W, c \leq C\}$. Пусть задано трехканальное изображение как частный случай тензора: $I = \{a_{ijk}\}, a_{ijk} \in \mathbb{R}^3, i \leq H, j \leq W$. И задана функция, возвращающая размер произвольного тензора:

$$F_{dim}: \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{N}^3,$$

$$F_{dim}(T) = (H, W, C)$$

Пример – для изображения I размером 640×480 и тремя каналами $F_{dim}(I) = (640, 480, 3)$. Также, пусть задано ядро свертки $K = \{k_{ijl}\}, k \in \mathbb{R}, F_{dim}(K) = (H^*, W^*, C^*)$. Сверткой тензора T с ядром K называется функция:

$$F_{conv}(T, K)_{x,y} = \sum_{i=1}^H \sum_{j=1}^W \sum_{l=1}^C k_{ijl} T_{x+i-1, y+j-1, l}$$

Для уменьшения размера тензора используется функция субдискретизации: $F_{pool}: \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H' \times W' \times C}$, $H' < H, W' < W$. Для добавления нелинейности используется функция активации:

$$F_{act}(T)_{x,y,z} = f(p_{h,w,c}),$$

где $f(p_{h,w,c})$ - нелинейная функция (например, \arctg , \tanh , сигмоида и т.д.).

Блоком кодировщика называется последовательность $F_{act}(F_{pool}(F_{conv1} \circ \dots \circ F_{convN}(T, K))) = T', T'$ - набор особенностей, \circ - операция композиции.

В работе используется ResNet50, который является одним из самых эффективных кодировщиков для выделения набора особенностей изображений [15]. Несмотря на то, что архитектура является достаточно глубокой (16 блоков) с несколькими остаточными (residual)[105] блоками, ее быстродействие хватает для работы в реальном времени. Остаточным блоком называется поэлементное сложение поступающего на вход тензора с тензором, получившимся на выходе сверки. Выходом стандартной архитектуры ResNet для изображения размером $640 \times 480 \times 3$ является набор особенностей размера $20 \times 15 \times 2048$. Обозначим эту архитектуру как **Basic**.

Была исследована облегченная версия архитектуры ResNet50 без последних трех блоков. Выход сети имеет размер $30 \times 40 \times 1024$. Архитектура работает гораздо быстрее исходной, при этом предполагается, что потери в точности будут незначительные. Облегченная версия архитектуры обозначена как **Lite Basic**.

Декодировщик. Декодировщик предназначен для восстановления одноканальной карты глубины из полученного кодированного представления изображения. Для повышения частоты дискретизации набора особенностей, были исследованы алгоритмы, позволяющие ускорить обработку изображений при сохранении точности на подходящем для решения задачи mvSLAM уровне. Один блок декодировщика, как правило, состоит из комбинации операций повышения частоты дискретизации $F_{upsample}: \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H' \times W' \times C}$, $H' > H, W' > W$, свертки

F_{conv} и активации F_{act} . В результате работы декодировщика получается карта глубины $D = \{d_{ijk}\}$, $i \leq H$, $j \leq W$, $k = 1$.

Первая модификация декодировщика состоит из разверточных (**Deconv**) слоев для преобразования набора особенностей изображения в карту глубины. Т.е. в каждом блоке вместо $F_{upsample}$ и F_{conv} используется обратная функция свертки F_{conv}^{-1} . Блок выглядит следующим образом: $F_{act}(F_{conv}^{-1}(T, K))$. Такой подход является базовым для большинства современных декодировщиков, применимых к задаче восстановления глубины, и он был использован как основание для сравнения предложенных модификация.

Вторая модификация содержит слой повышения дискретизации с последующий серией одномерных сверток (non-bottleneck 1D convolution), которая использовалась ранее в сети ERFNet[76]. Повышение дискретизации происходит методом интерполяции ближайшего соседа, затем следуют два последовательных набора сверток с ядром размера 3×1 и 1×3 и выполняется нормализация (batch normalization). Блок такой архитектуры кодировщика выглядит следующим образом: $F_{act}(F_{norm}(F_{int}(F_{conv}(F_{conv}(T, K_1), K_2))))$ Выход активируется функцией ReLU. Модификация обозначена как **Upsampling + nonbt**.

Третья модификация – использование слоев сверточного повышения дискретизации (**up-convolution**)[56] в декодировщике. Декодировщик состоит из серии операций повышения дискретизации и сверток: $F_{act}(F_{conv}(F_{pool}^{-1}(T), K))$. Подход работает быстрее, чем **deconv** при сравнимой точности. Исследована операция чередования (**interl**) для увеличения быстродействия.

Проекции (shortcuts). В работе использованы проекции (обозначены как **SC**) из слоев кодировщика в декодировщик. В рассматриваемом случае, проекцией является поэлементное суммирование двух тензоров. Такой подход позволяет повысить точность восстановления глубины на границах объектов.

3.2 Алгоритмические оптимизации разработанных моделей искусственных нейронных сетей

Для уменьшения времени обработки изображения было предложено заменить блок **сверточного повышения дискретизации** в декодировщике на эквивалентную реализацию с помощью функции **чередования**. Основная идея состоит в том, чтобы заменить медленные операции повышения дискретизации и последующую свертку на серию из четырех сверток меньших размеров и операцию чередования, которая используется в формате кодировки jpeg. Таким образом удастся сократить время обработки изображения без потери качества. Также, операция чередования поддается распараллеливанию для вычисления на графическом процессоре.

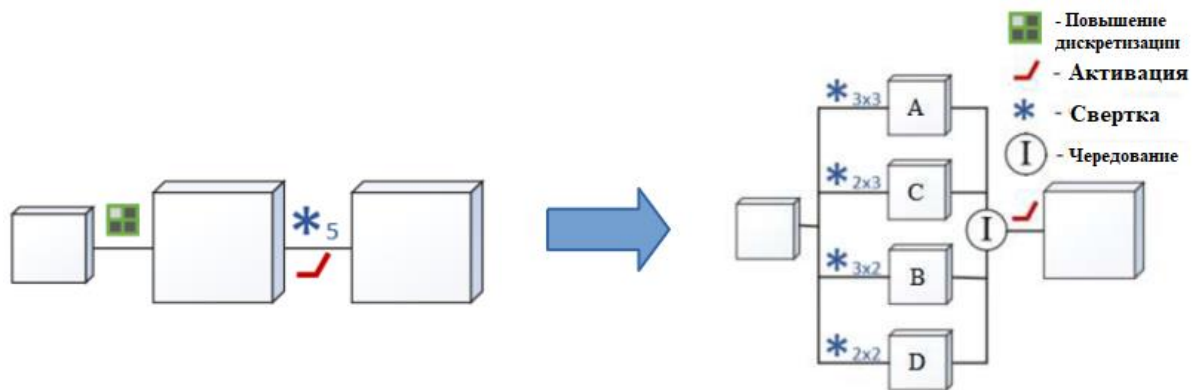


Рис. 21. Замена операции сверточного повышения дискретизации на чередование. (Слева – блок сверточного повышения дискретизации, который состоит из операций повышения частоты дискретизации, свертки с ядром 5x5 и активации. Справа – эквивалентная замена из серии сверток, операции объединения весов чередованием и активации).

Сама реализация блока чередования заключается в следующем. На вход алгоритму подается четыре четырехмерных тензора размером $N \times H \times W \times C$, полученных путем последовательной свертки исходного тензора с разбитыми

ядрами (см. Рис. 22). Для каждого элемента вычисляется, является ли его текущая позиция по H и W четной или нечетной. В зависимости от четности/нечетности позиций, текущий элемент занимает свою позицию в новом тензоре размером $N \times 4H \times 4W \times C$.

Графически объяснение замены свертки 5×5 операцией чередования представлено на Рис. 22. Исходное ядро свертки размера 5×5 [93], которое используется в блоке декодировщика up-convolution, разбивается на 4 свертки меньших размеров (3×3 , 2×3 , 3×2 и 2×2). Значения с нечетными строками и столбцами переносятся в первую свертку, с четными строками, нечетными столбцами – во вторую, нечетными строками, четными столбцами – третью, четными строками и столбцами – четвертую.

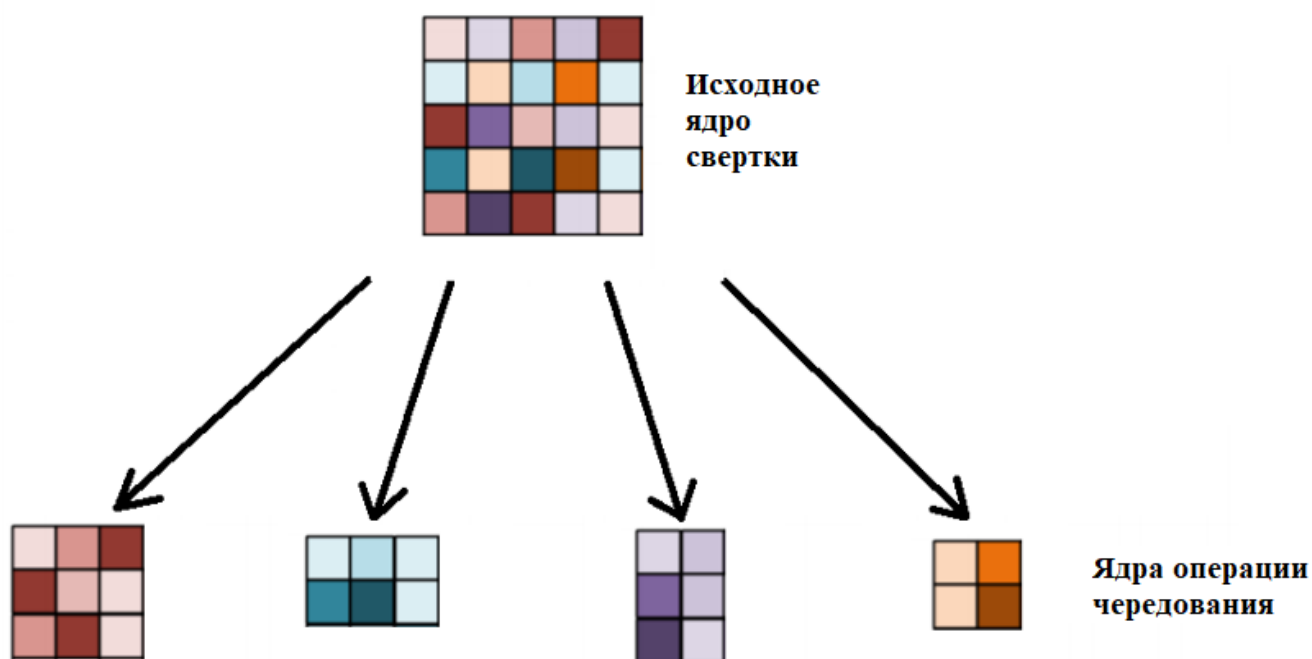


Рис. 22. Иллюстрация замены свертки на операцию чередования.

Реализация алгоритма чередования:

1. **Вход:** A, B, C, D – четырехмерные тензоры размера (N, H, W, C) , N – количество тензоров в пачке, H – высота изображений, W – ширина изображений, C – количество каналов изображений

```

2. Выход: Out – четырехмерный тензор размера (N,4H,4W,C)
3. for всех  $i < N * H * W * C$  do
4.    $n_{in} = i / (H * W * C);$ 
5.    $h_{in} = (i \bmod (H * W * C)) / (W * C);$ 
6.    $w_{in} = (i \bmod (W * C)) / C;$ 
7.    $index_{in} = n_{in} * H * W * C / 4 + (h_{in}/2) * W * C/2 + (w_{in}/2) * C + c_{in};$ 
8.   if  $h_{in}$  – четное then
9.     if  $w_{in}$  – четное then
10.      Out[i] = A[indexin];
11.    else
12.      Out[i] = B[indexin];
13.    end
14.  else
15.    if  $w_{in}$  – четное then
16.      Out[i] = C[indexin];
17.    else
18.      Out[i] = D[indexin];
19.    end
20.  end
21. end

```

Однако, у такого подхода существует значительный недостаток – архитектуры нейронных сетей с блоком **чередования** в декодировщике обучаются гораздо медленней, чем архитектуры с оригинальным блоком. Поскольку обе операции (up-convolution и свертки с последующим чередованием) являются эквивалентными, то было предложено использовать операцию **сверточного повышения дискретизации** для обучения сети, а затем обученные веса переносить в модификацию сети с операцией чередования (**interleaving**). Таким

образом удастся получить прирост производительности без потери качества восстановления глубины.

3.3 Функции потерь.

В процессе обучения разработанных архитектур искусственных нейронных сетей, было установлено, что ошибка предсказанной глубины на близких расстояниях (менее двух метров) больше чем средняя ошибка по всей предсказанной карте глубины. Большая ошибка на ближних расстояниях (см. Рис. 23) значительно влияет на качество навигации сложной робототехнической системы, поскольку такие глубины по сути являются тем, что находится ближе всего к роботу. Т.е. глубины на ближних расстояниях играют ключевую роль.

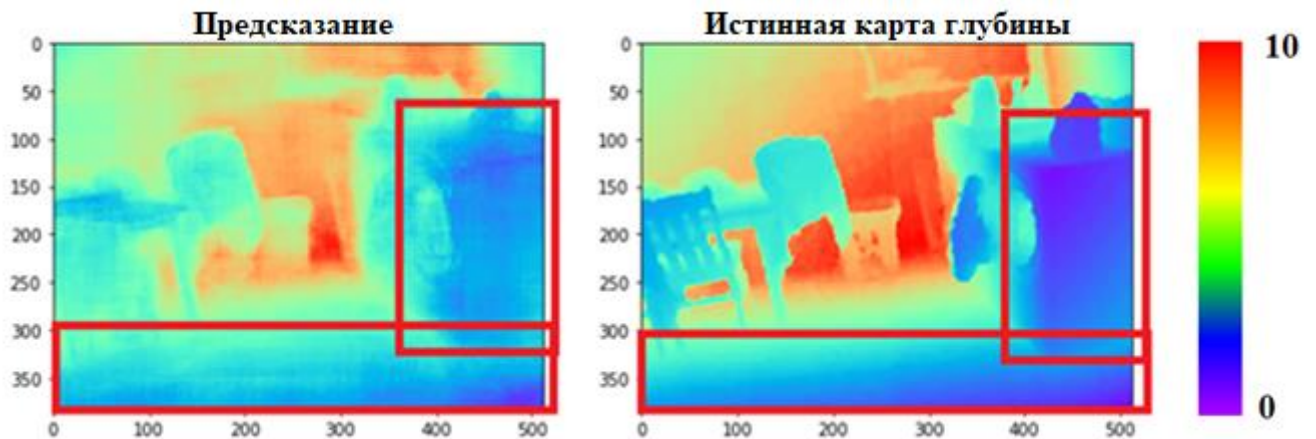


Рис. 23. Иллюстрация проблемы восстановления глубин на близких дистанциях. Красными прямоугольниками выделены области, на которых предсказанные глубины сильно отличаются от истинных.

Стандартным подходом к решению этой проблемы является обучение с использованием функции потерь **BerHu**:

$$L(D_{ij}, D_{ij}^*) = \begin{cases} |D_{ij}^* - D_{ij}|, & |D_{ij}^* - D_{ij}| < k, \\ \frac{(D_{ij}^* - D_{ij})^2 + k^2}{2k}, & |D_{ij}^* - D_{ij}| \geq k. \end{cases}$$

$$BerHu = \frac{1}{H * W} \sum_{i=1}^H \sum_{j=1}^W L(D_{ij}, D_{ij}^*),$$

Где D_{ij} - истинная глубина, D_{ij}^* - предсказанная глубина, H – высота карты глубины, W – ширина карты глубины, k – пороговое значение (константа). Т.е. в явном виде задается порог, ниже которого значения штрафуются сильнее, чем значения выше порога. Однако, основной проблемой этой функции потерь является то, что пороговое значение k необходимо задавать вручную, что негативно сказывается на стабильности обучения на разных коллекциях. Целесообразно подбирать коэффициент k автоматически в процессе обучения.

В предложенной модификации функции **aBerHu** предлагается устанавливать пороговое значения исходя из значений эталонных глубин в обучающей коллекции данных:

$$L'(D_{ij}, D_{ij}^*) = \begin{cases} |D_{ij}^* - D_{ij}|, & D_{ij} < k, \\ \frac{(D_{ij}^* - D_{ij})^2 + k^2}{2k}, & D_{ij} \geq k. \end{cases}$$

$$aBerHu = \frac{1}{H * W} \sum_{i=1}^H \sum_{j=1}^W L'(D_{ij}, D_{ij}^*)$$

Таким образом, был предложен адаптивный подход для вычисления порогового значения (k) в таких функциях. На каждом шаге обучения определяется окно $[k - \delta; k]$ и $[k + \delta; k]$, δ – половина длины окна (задается вручную). Для всех $D_{ij} \in [k - \delta; k]$ вычисляется сумма $R_{lower} = \sum |D_{ij} - D_{ij}^*|$. Аналогичный шаг делается для $D_{ij} \in [k + \delta; k]$. Сумма ошибок записывается в R_{upper} . Далее, значения R_{lower} сравнивается с R_{upper} , и обновляется пороговое значение k :

$$k = \begin{cases} k - lr * \delta, & R_{lower} < R_{upper} \\ k + lr * \delta, & R_{lower} > R_{upper} \end{cases}$$

lr – гиперпараметр, коэффициент чувствительности обучения (задается вручную для процесса обратного распространения ошибки).

Также, на ряду с разработанной функцией потерь, было предложено использовать взвешенную сумму стандартных функций, используемых при обучении искусственных нейронных сетей, а именно среднеквадратичную ошибку **MSE** и относительную ошибку **REL**. Такой подход позволит уменьшить время обучения, поскольку обе функции вычисляются быстро, а также позволит повлиять на описанную проблему близких расстояний.

$$F_{mse+rel}(D_{ij}, D_{ij}^*) = \alpha_1 * MSE(D_{ij}, D_{ij}^*) + \alpha_2 * REL(D_{ij}, D_{ij}^*),$$

$$MSE(D_{ij}, D_{ij}^*) = \frac{1}{H * W} \sum_{i=1}^H \sum_{j=1}^W (D_{ij}^* - D_{ij})^2$$

$$REL(D_{ij}, D_{ij}^*) = \frac{1}{H * W} \sum_{i=1}^H \sum_{j=1}^W \left(1 - \frac{D_{ij}}{D_{ij}^*}\right)^2$$

α_1, α_2 - настраиваемые параметры, которые указывают на то, какая функция ошибки будет больше влиять на финальный результат обучения: абсолютная или относительная.

3.4 Исследование программной реализации разработанных моделей на встраиваемой платформе

Разработанные архитектуры были исследованы на встраиваемой платформе NVidia Jetson TX2. Архитектуры были реализованы с помощью фреймворков Tensorflow и Keras на языке Python. Операции чередования (**interleaving**), а также сверточного повышения дискретизации (**up-convolution**) были реализованы для CPU и GPU на языке C/C++ с помощью компиляторов g++ и nvcc соответственно. Для обучения был использован гибридный высокопроизводительный вычислительный комплекс ФИЦ ИУ РАН [110].

На этапе обучения и валидации была использована открытая коллекция данных NYU Dataset v2, которая состоит из более чем 400 000 пар изображение-глубина, заснятых на более 470 разных сценах. Для увеличения разнообразия изображений из NYU Dataset v2 были случайно обрезаны (вместе с соответствующими ими картами глубин), а также случайно отображены и повернуты.

Результаты тестирования описанных в статье архитектур представлено в таблице 5. В таблице представлены как данные, полученные для разработанных архитектур, так и для известных. Поле Loss указывает на функцию потерь, которая была использована при обучении архитектуры, Input resolution – размер входящего изображения, Decoder – используемый вариант декодировщика, MSE – значение среднеквадратичной ошибки в метрах (меньше – лучше), REL – значение относительной ошибки в метрах (меньше – лучше), δ^i – дельта метрика (больше – лучше), PC time – время обработки одного изображения в секундах (меньше – лучше), Jetson time – время обработки одного кадра на встраиваемой платформе Nvidia Jetson TX2 в секундах (меньше – лучше). Результаты тестирования архитектур восстановления глубины изображений

Таблица 6. Результаты экспериментального исследования разработанных архитектур искусственных нейронных сетей.

	Loss	Input resolution	Decoder	MSE	REL	δ^1	δ^2	δ^3	PC time (s)	Jetson time (s)
Wang et al.	Custom	-	-	0.555	0.220	0.605	0.890	0.970	-	-
Eigen et al.	Custom	304x228	-	0.823	0.215	0.611	0.887	0.971	-	-
Laina et al.	BerHu	304x228	-	0.328	0.127	0.811	0.953	0.988	-	-
Alhashim et al.	Custom	640x480	-	0.152	0.103	0.895	0.980	0.996	-	-
Basic	BerHu	640x480	Deconv	0.467	0.186	0.718	0.929	0.980	0.144	0.152
Basic + SC	BerHu	640x480	Deconv	0.487	0.194	0.695	0.915	0.975	0.521	0.563
Basic + SC	aBerHu	640x480	Upsampling + nonbt	0.440	0.184	0.725	0.932	0.982	0.158	0.215
Basic + SC	MSE + REL	640x480	Upsampling + nonbt	0.419	0.173	0.748	0.944	0.987	0.158	0.215
Basic + SC	MSE + REL	320x240	Upsampling + nonbt	0.408	0.180	0.746	0.940	0.984	0.049	0.062
Lite Basic + SC	MSE + REL	320x240	Upsampling + nonbt	0.533	0.202	0.687	0.915	0.979	0.035	0.049
Basic + SC + interl	MSE + REL	640x480	Up-convolution	0.514	0.206	0.708	0.912	0.970	0.285	0.328
Basic + SC + interl	MSE + REL	320x240	Up-convolution	0.580	0.215	0.673	0.899	0.965	0.057	0.067
Basic + SC + interl + T	MSE + REL	640x480	Up-convolution	0.445	0.178	0.714	0.939	0.987	0.181	0.227
Basic + SC + interl + T	MSE + REL	320x240	Up-convolution	0.495	0.181	0.724	0.940	0.983	0.048	0.061
Lite Basic + interl + T	MSE + REL	640x480	Up-convolution	0.658	0.233	0.642	0.886	0.964	0.101	0.135
Lite Basic + interl + T	MSE + REL	320x240	Up-convolution	0.660	0.236	0.649	0.881	0.960	0.027	0.037

Качественно результаты восстановления глубины изображений можно оценить на Рис. 24.

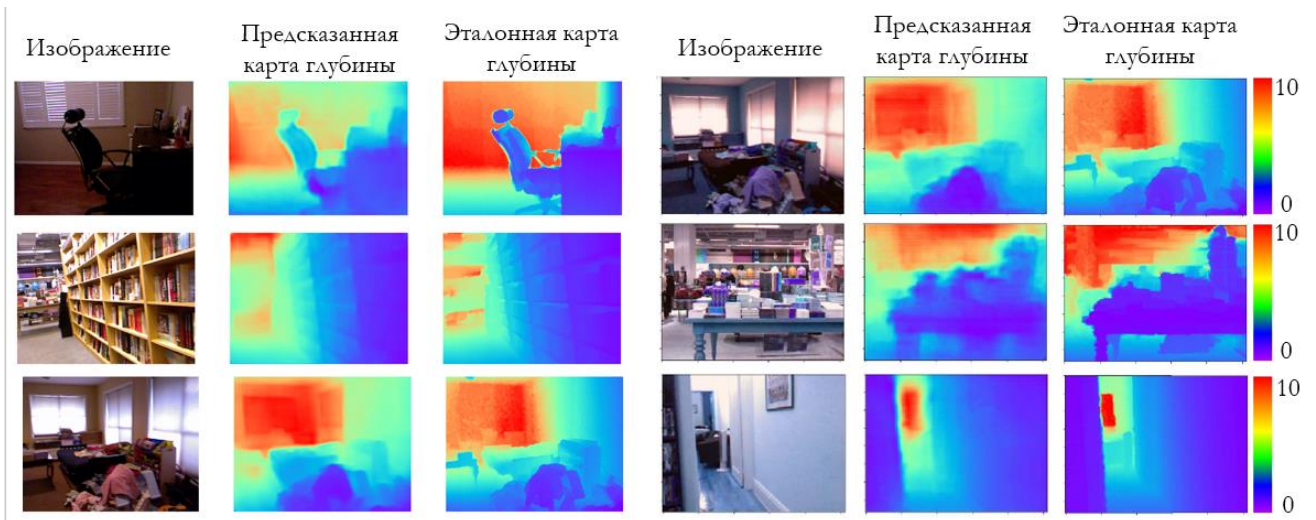


Рис. 24. Качество карт глубин, восстановленных архитектурой **Lite Basic + SC** на изображениях коллекции NYU Dataset v2.

Лучший результат в плане быстродействия на NVidia Jetson TX2 показала архитектура Lite Basic + interl + T (37 мс). Время обработки сравнимо с методом, представленным в [61], при этом отмечается незначительная для задачи mvSLAM потеря в качестве.

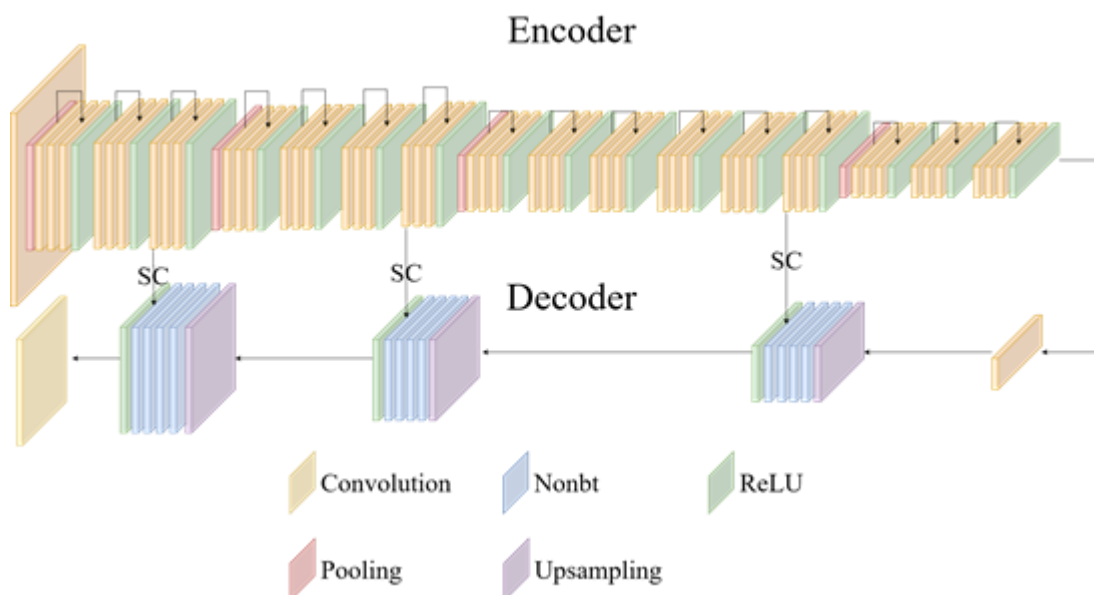


Рис. 25. Реализованная на встраиваемой платформе архитектура искусственной нейронной сети.

С другой стороны, в архитектурах Basic + SC + Upsampling + nonbt удалось добиться лучшего показателя REL, чем в некоторых архитектурах из [22], которые не нацелены на работу в реальном времени. При этом, один кадр обрабатывается 62мс, что достаточно для использования в задаче mvSLAM.

Архитектура Lite Basic дает большой прирост к скорости работы, но при этом теряет в точности, даже с переносом знаний с более тяжелой архитектуры. Потеря в точности может быть компенсирована модификациями, представленными в статье (Upsampling + nonbt, Up-convolution) и ускорены с помощью операции interl.

Экспериментальное исследование показало, что разработанные архитектуры позволяют производить восстановление глубины изображений в реальном времени как на удаленной станции, так и на современном бортовом вычислителе малой робототехнической системы. При этом качество полученных архитектур сопоставимо с актуальными решениями в области восстановления глубины изображений.

3.5 Реализация в составе программного комплекса картирования и локализации

Для реализации системы одновременного картирования и локализации по видеопотоку для малых робототехнических систем, целесообразно использовать широко применяемый для робототехнических систем программный комплекс Robot Operating System (ROS). ROS предоставляет внутренний механизм взаимодействия отдельных узлов (nodes) посредством специальных каналов для передачи данных (topics). Узлы исполняются независимо друг от друга, и каждый из них выполняет отдельную задачу. Существует большое количество готовых открытых узлов для решения широкого круга исследовательских задач, в частности, для решения задачи mvSLAM. Одним из таких решений является RTABMAP [54]. RTABMAP представляет из себя узел ROS, который принимает на

вход видеопоток (или набор изображений) и набор карт глубин для каждого изображения. На выходе получается карта местности, в которой производилась съемка, в виде облака точек и набор позиций камеры в полученной карте.

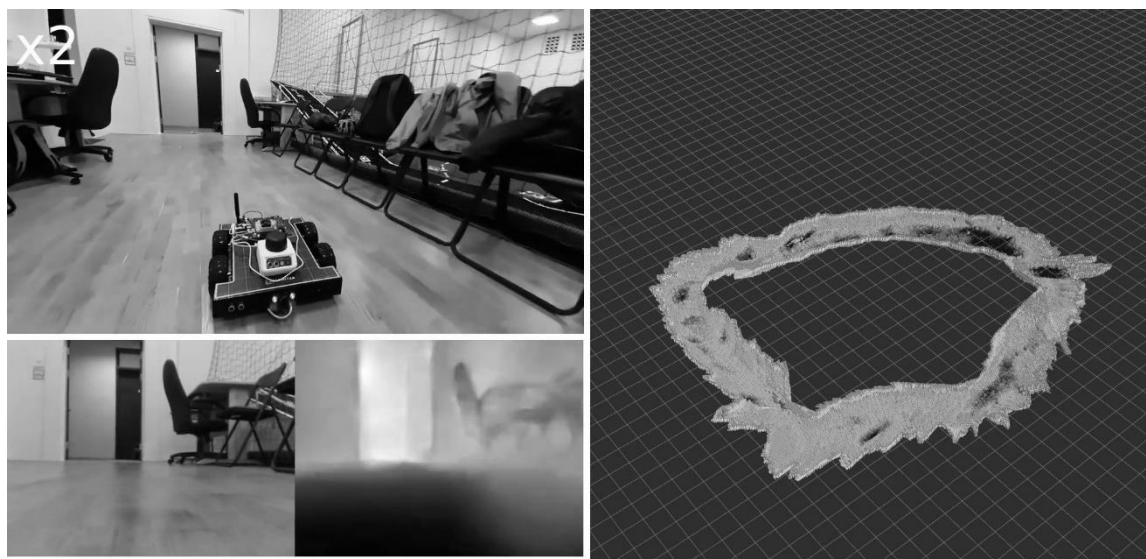


Рис. 26. Пример работы полученной системы на примере реальной робототехнической системы

Для реализации системы был разработан узел, отвечающий за реконструкцию глубины с помощью изображения, полученному с единственной камеры и предусмотрена возможность использования обученных весов моделей искусственных нейронных сетей. Программная реализация сверточной нейронной сети была разработана с использованием набора открытых программных средств TensorFlow, которые обеспечивают возможность обработки изображений и обучение сети на графическом процессоре, что повысит скорость работы алгоритмов обучения и выделения глубин изображений.

Работа системы организована следующим образом: от камеры робототехнической системы на бортовой вычислитель поступает последовательность цветных изображений (видеопоток). Узел «UV Driver» принимает изображение и преобразует его в формат `sensor_msgs/Image` и передает его через топик `image_raw` следующему узлу. «CNN Image Depth Reconstruction Node» использует полученные на этапе обучения веса сверточной нейронной сети

(подробнее об этом - ниже) для восстановления карты глубины изображения. Полученная карта глубины передается через топик `depth_image` следующему узлу «RATVMAP», который производит картирование и локализацию с использованием изображения и его карты глубины. На выходе системы – восстановленная карта (облако точек) `pointcloud` и позиция камеры в полученной карте `camera_pose`.

Полученная система была протестирована на робототехнической системе МПРМ Зарница в помещении робототехнического центра ФИЦ ИУ РАН. Робототехническая система двигалась по прямоугольной траектории 7x15 метров из помещения в коридор и обратно. Всего было произведено 20 запусков из разных начальных точек. Пример работы системы представлен на Рис. 26. На всех запусках алгоритм картирования и локализации допускал ошибки при повороте в низкодетализированной среде (коридор с однотонными стенами), однако эти ошибки были скорректированы алгоритмом замыкания траекторий по ходу движения МПРМ.

Все вычисления выполняются автономно на бортовом вычислителе NVidia Jetson TX2. Эксперименты показали, что разработанные архитектуры позволяют точно и быстро производить восстановление карты и вычисление позиций системы на встраиваемой платформе. FCNN работает на графическом процессоре со скоростью 16 кадров в секунду, в то время как алгоритм, решающий задачу mvSLAM, – на центральном, поэтому алгоритмы не занимают ресурсы друг у друга.

ROS позволяет выполнять картирование и локализацию в двух сценариях: 1) нейронная сеть работает на NVidia Jetson TX2, затем результаты отсылаются на рабочую станцию и vSLAM выполняется уже там, 2) оба алгоритма работают на NVidia Jetson TX2. Оба сценария подразумевают выполнение в реальном времени. Код всех исследований и реализация для TensorRT и ROS доступны по ссылке: <https://github.com/cnndepth>. Видео с демонстрацией результата доступно по ссылке: <https://www.youtube.com/watch?v=ayjvfzm-C7s>.

3.6 Выводы по главе

Одной из основных проблем известных методов, решающих задачу одновременного картирования и локализации по видеопотоку, является низкая детализация результирующих карт. Для решения проблемы низкой детализации целесообразным представляется использование восстановленных карт глубин изображений в качестве входных данных. Таким образом задача mvSLAM сводится к задаче RGBD-SLAM, для которой известны методы, обладающие высокой точностью и эффективностью. Однако, известные методы восстановления карт глубин изображений опираются на использование нейросетевых моделей с большим числом параметров, использование которых на применяемых на борту мобильного робота вычислителях в реальном времени не представляется возможным. Для устранения этой проблемы в работе была предложена нейросетевая модель, обладающая, с одной стороны, высокой вычислительной эффективностью, а с другой – высокой точностью предсказания глубин изображений.

Разработанная модель представляет собой полносверточную искусственной нейронную сеть, состоящую из кодировщика и декодировщика. Для достижения работы в реальном времени была применена стандартная архитектура ResNet50 в качестве кодировщика, и операции повышения дискретизации с nonb1-свертками в декодировщике, а также разработаны функции потерь для увеличения точности предсказания на этапе обучения. Разработанная модель была исследована на вычислителе NVidia Jetson TX2, который часто применяется в мобильной робототехнике, с применением открытой коллекции NYU Dataset v2. Для количественного сравнения, были использованы открытые реализации современных искусственных нейронных сетей, обученных на той же коллекции, что и разработанные в диссертационном исследовании модели. Исследование показало, что предложенная модель может работать в реальном времени (обработка более 16 кадров в секунду для изображений с разрешением 320x240

пикселей) при сравнимой с современными архитектурами точностью. Далее был разработан программный компонент восстановления глубины изображения для операционной системы роботов (Robotic operating system, ROS) и включен в состав системы картирования и локализации на основе алгоритма RTAB-Map. Полученный программный комплекс был экспериментально протестирован на малом мобильном роботе. Эксперимент состоял в проезде робота по коридору робототехнического центра ФИЦ ИУ РАН с запущенными разработанными алгоритмами предсказания глубины и RTAB-Map, при этом по мере движения не должно было происходить потерь локализации. Тестирование показало, что архитектуры могут быть использованы автономно на встраиваемой системе для картирования и локализации по видеопотоку, а в результате была получена детализированная карта коридора робототехнического центра ФИЦ ИУ РАН.

Глава 4. Разработка моделей для оценки результатов картирования и локализации, полученных методами mvSLAM

В настоящее время известно множество методов, решающих задачу mvSLAM, использующих разные подходы к построению карты и локализации. Как правило, выходными данными этих методов является траектория перемещения камеры в виде последовательности её позиций в каждый момент времени в некоторой глобальной системе координат и карта окружающей среды в формате облака точек (Point cloud) [78] или OctoMap[100]. Очевидно, что для анализа эффективности картирования и локализации требуется оценка качества работы этих методов, т.е. оценка того, насколько вычисленные траектория и карта отличаются от действительных (эталонных). Это нетривиальная техническая задача, т.к. получение эталонных траектории и карты может быть затруднительно. Сложность решения задачи зависит от того, какая методика тестирования используется. Можно выделить три класса подобных методик: с использованием реальных робототехнических систем в реальных средах, с использованием заранее подготовленных и общедоступных коллекций данных, с использованием специальных симуляционных сред.

При тестировании методов, решающих задачу mvSLAM, на реальных робототехнических системах количество возможных повторений эксперимента обычно ограничено, и они сложно воспроизводимы. Более того, из полученных по результатам экспериментов данных, не ясно насколько рассматриваемый метод применим на других роботах и/или в других условиях.

Для решения проблемы воспроизводимости могут быть использованы открытые коллекции данных, в которых представлены траектории робота в определенном помещении с показаниями сенсоров, например коллекция EuRoC [11]. Зачастую, но не всегда, в коллекциях помимо показаний сенсоров в каждый момент времени представлены эталонные траектории и карты, что позволяет проводить сравнительный анализ нескольких алгоритмов. Недостатком

использования подобных коллекций является их ограниченность: сравнение возможно только по имеющимся в коллекции траекториям.

С развитием вычислительной техники получили распространение симуляторы, подходящие для проведения экспериментов и сравнения разрабатываемых методов и алгоритмов для решения задачи mvSLAM. Одним из наиболее известных симуляторов является Gazebo [52,65]. Подобные симуляторы обычно поддерживают возможность создания собственных моделей помещений и роботов, получения данных с установленных на роботе сенсоров и истинных координат всех объектов. К недостаткам использования симуляторов можно отнести тот факт, что в них неизбежно присутствуют искажения, связанные с невозможностью точного моделирования физических процессов (распространение света, движение робота, работа сенсоров и т.д.). Однако, несмотря на подобные искажения, симуляционные среды являются важным инструментом для оценки качества методов. Так только в симуляторе можно провести эксперименты на большом количестве произвольных помещений с известными истинными траекториями и картами.

Помимо метрик, необходимо разработать среду, в которой возможно эти метрики вычислить. Современные робототехнические симуляторы дают возможность проводить эксперименты на большом количестве разнообразных помещений с произвольной геометрией и интерьером. В отличие от коллекций данных, созданных по реальным помещениям, в симуляционной среде можно выбрать произвольную сцену и траекторию для тестирования. Большое число экспериментов позволяет точнее оценить качество алгоритмов. В данной работе рассматривается два робототехнических симулятора - Gazebo и Habitat.

Симулятор Gazebo является одним из самых популярных в робототехническом сообществе. Он интегрирован с ROS, и в нем есть множество моделей помещений и роботов, находящихся в открытом доступе. В Gazebo смоделированы многие физические процессы, включающие инерцию и гравитацию, влияние атмосферы и распространение света. Gazebo предоставляет

возможность создания моделей помещений из готовых объектов. Пример модели представлен на Рис. 27.

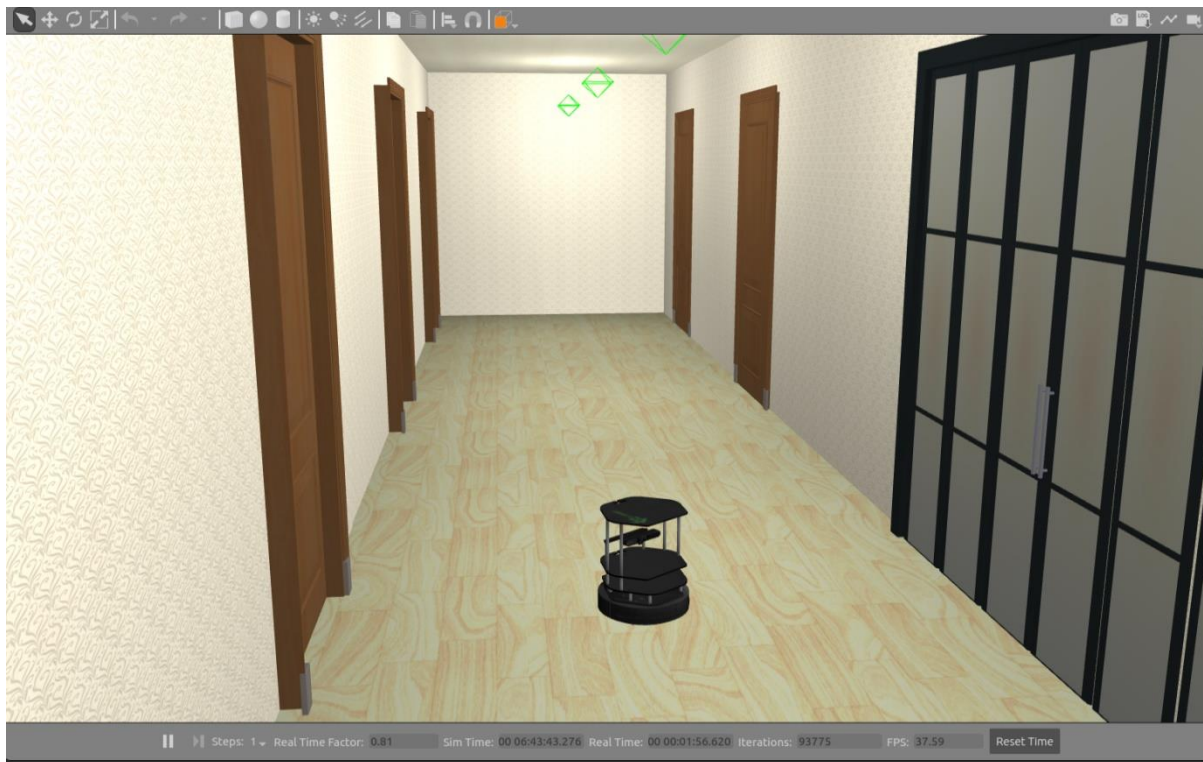


Рис. 27. Пример модели помещения в симуляторе Gazebo

Существенным недостатком тестирования алгоритмов vSLAM в симуляторе Gazebo является однообразие текстур в помещении, затрудняющая извлечение особых точек из изображений. Еще одним недостатком данного симулятора являются искажения распространения света, например, прохождение света сквозь непрозрачные объекты (см. Рис. 28). Эти недостатки приводят к существенным затруднением или невозможности исследования качества работы алгоритмов одновременного картирования и локализации по видеопотоку в симуляционной среде Gazebo. В связи с этим, целесообразным является использование симулятора, в котором перечисленные недостатки отсутствуют.

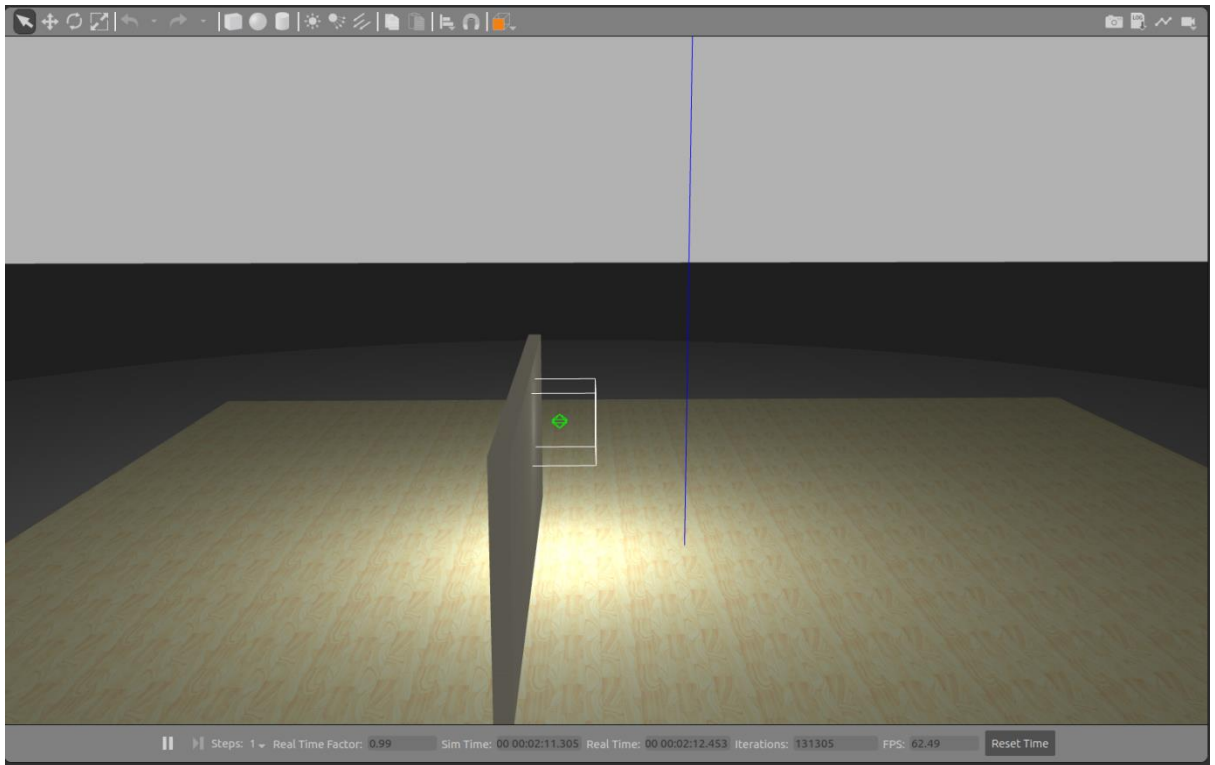


Рис. 28. Прохождение света сквозь стену в симуляторе Gazebo

Таким симулятором является Habitat, который был разработан в 2018 году для исследования алгоритмов обучения с подкреплением. В этом симуляторе поддерживаются коллекции Gibson [101] и Matterport3D [14], в которых представлены реалистичные модели помещений (см. Рис. 29), созданные по реальным помещениям с помощью датчиков глубины и motion-capture систем.

В симуляторе Habitat многие физические процессы смоделированы не полностью. В частности, движение робота дискретно, недоступны показания инерциальных навигационных систем. Это позволяет экономить вычислительные ресурсы, затрачиваемые на симуляцию, сохраняя возможность тестирования алгоритмов одновременного картирования и локализации по видеоданным.



Рис. 29. Примеры модели помещения из коллекции Matterport 3D в симуляторе Habitat

В диссертационной работе рассматривается задача оценки качества алгоритмов картирования и локализации в симуляционных средах. Описывается симулятор Habitat [79], который, в отличие от аналогов (например - Gazebo), характеризуется высокой фотореалистичностью модельной среды. Поскольку в данном симуляторе у пользователя изначально нет доступа к эталонной модели помещения, описывается процесс получения такого эталона. Предлагается новая метрика качества картирования и локализации, пригодная для использования в симуляционной среде и учитывающая не только ошибку восстановления карт и траекторий, но и порядок построения карты методами, решающими задачу mvSLAM. Приводится пример применения предложенной методики экспериментального исследования для оценки качества алгоритма RTAB-Map.

4.1 Оценка качества траектории

Для количественного сравнения методов, решающих задачу mvSLAM, необходимо использовать специальные метрики как для измерения точности построенной траектории, так и для качества результирующей карты. Для формализации задачи оценки качества алгоритмов предложена модель, которая состоит из двух модулей: модуля локализации и модуля картирования.

Измерение качества модуля локализации происходит независимо от модуля картирования. В случае задачи mvSLAM, результатом работы (выходными данными) модуля локализации является траектория, которая состоит из последовательности позиций робота (камеры) в среде. Каждой позиции соответствуют координаты в трехмерном пространстве и вектор направления. Для оценки качества полученных траекторий, в работе используются уже известные метрики: Absolute Trajectory Error (ATE)[107] и Relative Pose Error (RPE)[59].

Формально, проблема оценки качества работы модуля локализации может быть записана следующим образом. Пусть дана последовательность трехмерных координат или позиций:

$$V = \{v_t = (p_t, q_t), t \in [1; m], m \in N\}$$

где $p_t = (p_t^x, p_t^y, p_t^z)$ – позиция робота (камеры) в декартовой системе координат, $q_t = (q_t^x, q_t^y, q_t^z, q_t^w)$ – кватернион ориентации, t – момент времени. Заданы функции определения соответствия между истинными позициями и предсказанными: $f(p_t, q_t) = (p'_t, q'_t)$. Тогда метрики могут быть записаны следующим образом:

1) $ATE(V, f) = \sum_{t=1}^m \|p_t - f(p_t, 0)\|^2$ – сумма квадратов расстояний между позициями предсказанной и истинной траекториями V и V' соответственно;

2) $RPE(V, f) = \sum_{t=2}^m \|q_t^{-1} \otimes (p_t - p_{t-1}) - f(0, (q_t)^{-1}) \otimes (f(p_t, 0) - f(p_{t-1}, 0))\|^2$ – сумма расстояний между истинными и предсказанными смещениями на каждом шаге, q_t^{-1} – кватернион, приводящий направление оптической оси камеры в момент

времени t к вектору $(1,0,0)$, т.е. направляющий ее вдоль оси x , \otimes - операция вращения вектора по заданному кватерниону.

Обе метрики широко используются для оценки качества локализации. Метрика АТЕ позволяет оценить качество построения позиций на всей протяженности траектории, в то время как RPE оценивает смещения между соседними позициями. Проблема измерения качества траектории уже хорошо изучена, а представленные выше метрики позволяют сравнить качество модуля локализации с использованием единственной камеры с качеством локализации с использованием иных датчиков.

Обозначенные метрики можно рассчитать для различного набора данных. Рассмотрим распространённые способы организации открытых коллекций данных для оценки качества работы методов, решающих задачу mvSLAM.

4.2 Разработка метода оценки качества картоирования на основе модели выходных данных методов, решающих задачу mvSLAM

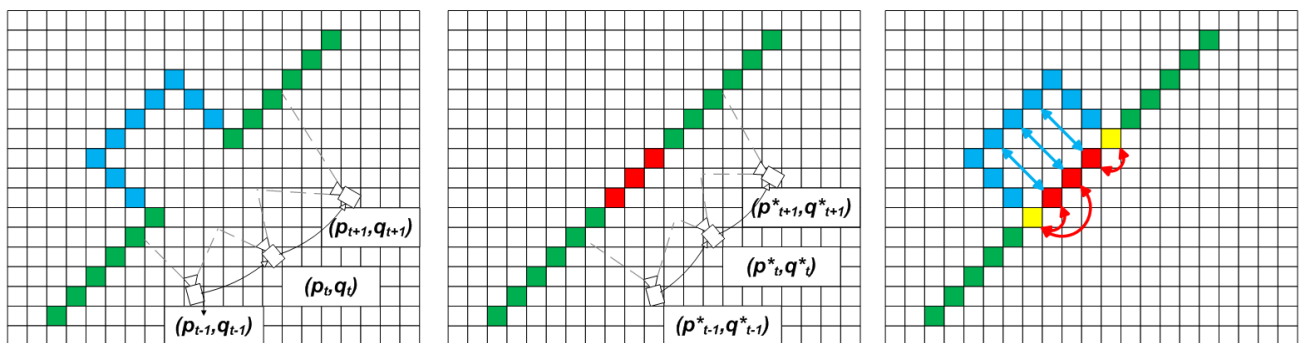


Рис. 30. Визуализация проблемы сравнения карт стандартными метриками. (А)

Показана истинная среда (зеленые и синие точки) и углы обзора камеры в разные моменты времени; Б) Построенная карта методами vSLAM (зеленые

и красные точки); В) Истинные соответствия между точками (синие стрелки) и вычисленные методом ближайшего соседа (оранжевые стрелки)).

Оценка качества модуля картирования методов, используемых для решения задачи mvSLAM, является более сложной задачей, по сравнению с оценкой качества построения траектории. Результатом работы модуля картирования, как правило, является множество трехмерных точек в пространстве, которое и представляет собой карту неизвестной местности. В связи с этим, возникает вопрос: как сравнить между собой два множества трехмерных точек, а конкретнее, как найти соответствия между точками полученного в результате решения задачи mvSLAM множества и истинного?

Для сопоставления точек используется алгоритм Iterative Closest Point (ICP) [8]. В качестве основных метрик можно использовать известные метрики, например среднеквадратичную ошибку (RMSE) [96] и относительную ошибку (REL) [86]. Метрики позволяют вычислять отклонение между соседними точками в двух множествах трехмерных точек. Однако для того, чтобы рассчитать метрику, найти соответствие между точками истинной карты и точками предсказанной карты.

Базовым подходом для нахождения соответствия точек двух множеств является поиск ближайших соседей после приведения двух множеств к единой системе координат. Однако, такой способ не подходит для оценки качества картирования методов, решающих задачу mvSLAM. В частности, одной из особенностей является частая ошибка при картировании дверных проемов в коридорной местности, когда сам проем наносится как препятствие на карту карту (Рис. 30. Б), в то время, как на этом месте должно быть пустое пространство с препятствием за ним (Рис. 30. А). При использовании сопоставления методом ближайших соседей, красные точки, нанесенные на карту будут сравниваться с желтыми точками, поскольку дистанция между ними – минимальна (на рисунке показано оранжевыми стрелками). Однако, красные точки должны быть нанесены

на место синих, которые принадлежат истинной карте и соответствие должно быть установлено между ними (синие стрелки на рисунке).

Вторая иллюстрация проблемы представлена на Рис. 31.

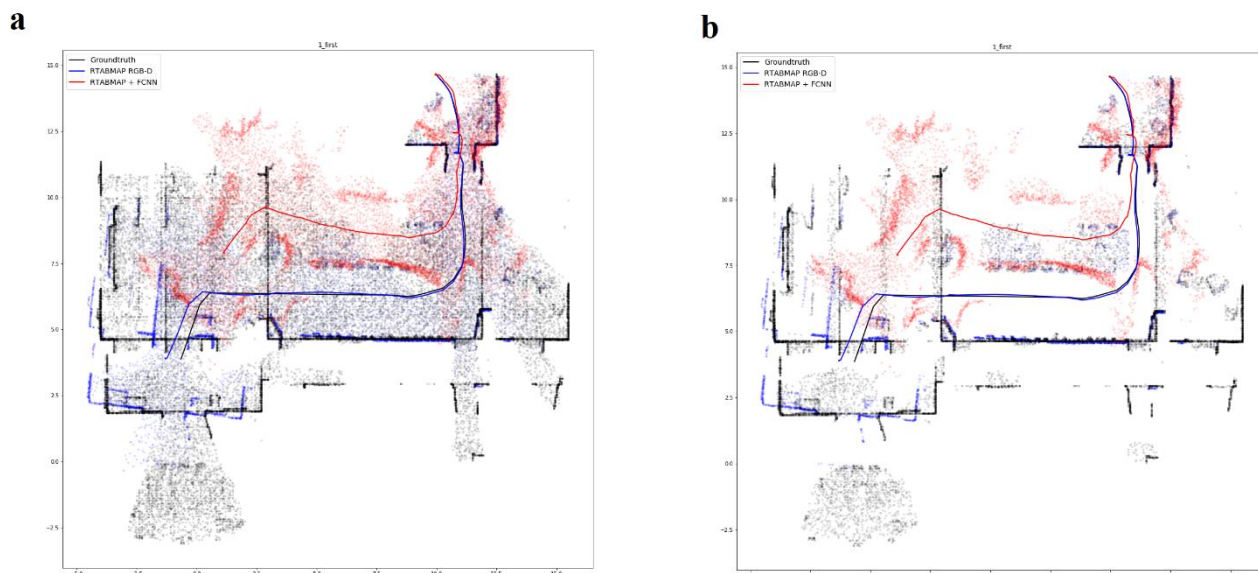


Рис. 31. Истинные карты и траектории, а также результаты работы методов vSLAM. а) Оригинальные результаты; б) Результаты с удаленным полом.

На рисунке представлены истинные траектория и карта (черная линия и точки), результаты, полученные методом RTAB-Мар с использованием истинных глубин (синяя линия и точки) и RTAB-Мар с использованием восстановленных искусственной нейронной сетью глубин (красная линия и точки). При расчете стандартных метрик сравнения облаков точек, происходит ситуация, при которой данные сравнения разнятся для оригинальных карт (а) и для карт с удаленным полом. Это обуславливается тем, что сравнение точек, которые принадлежат стенам на карте в оригинальной карте происходит с полом, поскольку они являются ближайшими. Если удалить пол, то ближайшими точками становятся уже другие стены, которые находятся дальше и поэтому значение метрики увеличивается. Однако, метрика для оценки качества картирования методов, решающих задачу vSLAM, должна быть инвариантна к подобного рода изменениям.

Описанные выше ситуации требуют разработки особого метода сопоставления точек двух множеств с учетом контекста задачи одновременного картирования и локализации по видеопотоку. Поэтому, в качестве «связующего» элемента между двумя множествами трехмерных точек, предлагается использовать позиции, которые использовались для реконструкции трехмерной карты из двумерного изображения. Для расчета метрики будет иметь значение не то, насколько близко находятся точки друг к другу, а то, где эти точки должны находиться в реальности в исследуемой местности. Реализация описанного подхода к сопоставлению точек реализована в метриках Absolute Mapping Error (AME) и Relative Mapping Error (RME), которые используют среднеквадратичное и относительное отклонение для количественного расчета качества карт.

Формально, поиск соответствий может быть записан следующим образом. Пусть даны истинная карта M и карта, построенная методом vSLAM карта M^* :

$$M = \{(x_i, y_i, z_i), i \in [1; n], x_i, y_i, z_i \in R, n \in N\}$$

$$M^* = \{(x^*_i, y^*_i, z^*_i), i \in [1; n], x^*_i, y^*_i, z^*_i \in R, n \in N\}$$

Также, пусть $\widehat{M}_t = \{m_i\} \subset M$ и $\widehat{M}_t^* = \{m_i^*\} \subset M^*$ - набор трехмерных истинных и рассчитанных точек, нанесенных на карту в момент времени t , p_t, q_t и p_t^*, q_t^* - позиции и ориентации истинные и рассчитанные соответственно.

Обозначим Q_t^* и Q_t матрицы поворота, определенные кватернионами q_t^* и q_t соответственно. Направление вектора $r_t = (Q_t^*)^{-1}Q_t(m_i^* - p_t^*)$ указывает на точку m_i^* на полученной в результате картирования карте. Вектор r_t получен путем нормализации векторной разности текущей позиции и конкретной точки из подмножества соответствующей позиции трехмерной области карты с последующим поворотом матрицей поворота истинной позиции. Т.е. после такой операции, истинная ориентация камеры совпадает с ориентацией камеры на предсказанной позиции. Точке m_t^* ставится в соответствие точка $m_i = p_t + \alpha r_t$, такая что:

$$\alpha = \min \{\alpha: p_t + \alpha r_t \in M_t\}$$

Т.е. функция соответствия выглядит следующим образом: $f(m_i^*) = p_t + \alpha r_t$. Соответствие ищется для всех точек из \widehat{M}_t и \widehat{M}_t^* . Пример расчета соответствия между точкой на истинной карте и точкой в предсказанной карте представлено на рис. 29.

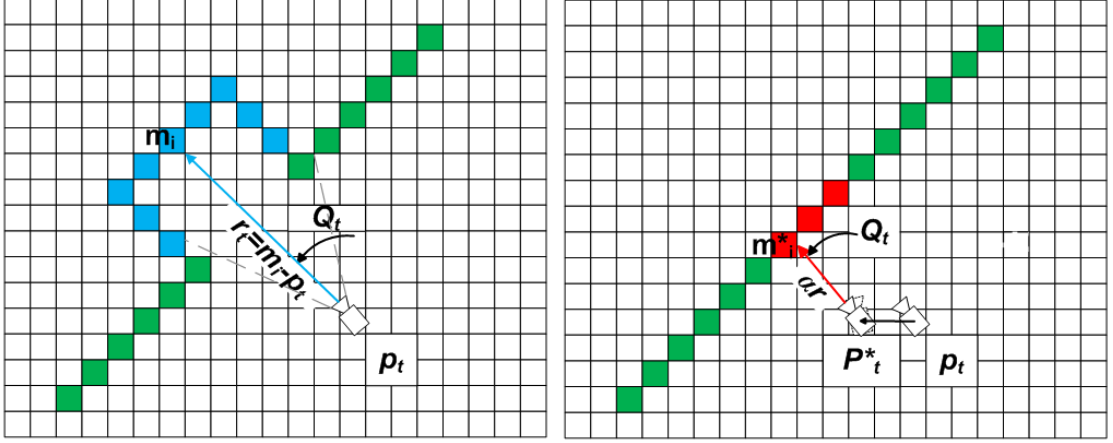


Рис. 32. Пример расчета соответствия.

Учитывая полученное соответствие, предлагается ввести следующие метрики:

- 1) $AME(M^*, f_{ME}) = \sqrt{\frac{1}{N} \sum_{t=1}^N \|f_{ME}(m_i^*) - m_i^*\|^2}$ - Absolute Mapping Error (AME), где $N = \min(|M|, |M^*|)$. Метрика рассчитывает среднее расстояние между точками, для которых установлено соответствие через r_t .
- 2) $RME(M^*, f_{ME}, f) = \sqrt{\frac{1}{n} \sum_{t=1}^N \|M_{q_t}(m_i - p_t) - M_{q_t^*}(m_i^* - p_t^*)\|^2}$ - Relative Mapping Error (RME). RME учитывает средние относительно текущей позиции расстояния до точек на карте с учетом установленного соответствия.

Таким образом, представленные метрики позволяют рассчитать количественные значения отклонения построенной карты от истинной с учетом представления о том, куда должны были быть нанесены конкретные элементы карты. Однако, при построении карты регулярно возникает ситуация, когда множество точек, нанесенных на карту с позиции (p_t, q_t) пересекаются с аналогичным множеством для $(p_{t+\delta}, q_{t+\delta})$, и тогда соответствия перестают быть

взаимно однозначными. Таким образом, финальные значения предложенных метрик AME и RME зависят от выбора момента времени t , при котором будет установлено соответствие между точками на истинной карте и vSLAM карте.

Возможны 4 варианта выбора параметра t :

1. t , при котором первый раз точка появилась в поле зрения карты (т.е. впервые была нанесена на карту);
2. t , при котором точка была видна в последний раз;
3. t , при котором точка находится ближе всего к камере;
4. Все t , при которых точка была нанесена на карту. При вычислении метрик AME и RME, дистанции между точками m_i^* и m_i , вычисленные с разных позиций, усредняются.

Первые три варианта чувствительны к изначальной дистанции до точки, а также к углу между камерой и точкой, что ведет к численному увеличению метрик AME и RME. 4-й вариант является устойчивым к вариации дистанции и угла, однако обладает большей вычислительной сложностью, нежели первые три варианта, поскольку пространство для поиска соответствий значительно увеличивается. Этот вариант используется в работе как основной.

4.3 Экспериментальное исследование в симуляционной среде

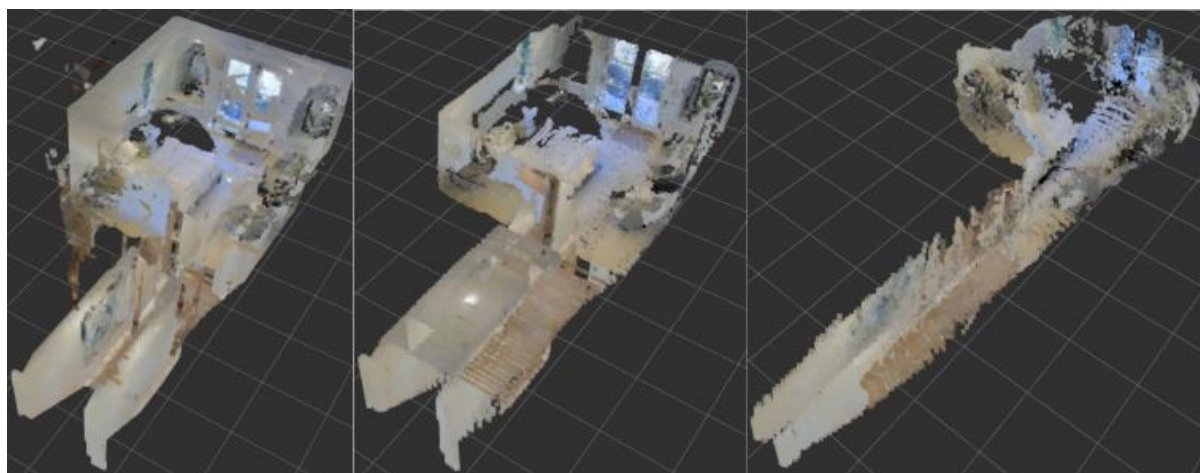


Рис. 33. vSLAM. Слева направо: истинная карта помещения (восстановлена методом, описанным в разделе 3), карта, построенная алгоритмом RTAB-Map по изображениям и глубинам из Habitat; карта, построенная алгоритмом RTAB-Map + FCNN.

В работе для тестирования алгоритмов картирования и локализации используется симулятор Habitat. Так как в данном симуляторе не представлен функционал для прямого получения точных координат объектов помещения, в данной работе был реализован метод восстановления этих координат по полученным из симулятора позициям камеры и картам глубины.

Трехмерная модель помещения восстанавливается последовательно по картам глубин изображений и позициям камеры. По каждому пикселю (h, w) в каждый момент времени t на карту наносится точка, проекция которой попала на изображение в пикселе (h, w) :

$M_{h,w,t} = p_t + d_{h,w,t} * r_{h,w,t}$ где p_t - положение камеры в пространстве в момент времени t ;

$d_{h,w,t}$ - глубина пикселя (h, w) изображения, полученного в момент времени t ;

$r_{h,w,t} = T * (1, \frac{w-W/2}{W/2}, \frac{h-H/2}{W/2})$ - направление, соответствующее пикселю (h, w) матрицы камеры в момент времени t ;

T - оператор вращения, переводящий вектор $(1, 0, 0)$ в направление оптической оси камеры в момент времени t .

В целях экономии памяти, необходимой для хранения карты, координаты всех точек округляются до 5 см.

Для эксперимента были отобраны 20 пар траекторий из симулятора Habitat (всего 40 траекторий и карт). Для каждой пары были получены истинные траектории и карты, траектории и карты, полученные алгоритмом RTAB-Map с использованием истинных глубин (RGB-D), и траектории и карты, полученные методом RTAB-Map и использованием восстановленных нейронной сетью глубин

(FCNN). Для каждой карты и траектории были рассчитаны ошибки локализации АТЕ, ошибка картирования с сопоставлением методом ближайшего соседа (AME) и ошибка картирования с разработанным методом сопоставления. Все метрики были рассчитаны относительно истинных карт и траекторий. Результаты экспериментов представлены в таблице 7.

Таблица 7. Результаты полунатурного исследования методов на оригинальной коллекции из симулятора Habitat

Карта	RGBD AME	FCNN AME	RGBD AME (our)	FCNN AME (our)	RGBD ATE	FCNN ATE	CC: detla	AME: delta	ATE: delta
1_first	0,139	0,441	0,524	2,808	0,26	2,506	3,173	5,359	9,638
1_second	0,11	0,474	0,493	3,285	0,226	2,319	4,309	6,663	10,261
2_first	0,455	0,607	1,413	2,741	0,895	2,453	1,334	1,94	2,741
2_second	0,116	0,495	0,636	3,353	0,31	2,699	4,267	5,272	8,706
3_first	0,031	0,251	0,185	0,792	0,04	0,474	8,097	4,281	11,85
3_second	0,113	0,265	0,363	0,951	0,198	0,573	2,345	2,62	2,894
4_first	0,335	0,773	1,355	2,87	1,033	2,24	2,307	2,118	2,168
4_second	0,127	0,484	0,439	3,426	0,284	2,444	3,811	7,804	8,606
5_first	0,052	0,312	0,209	1,704	0,095	1,34	6	8,153	14,105
5_second	0,1	0,267	0,36	1,237	0,126	0,831	2,67	3,436	6,595
6_first	0,168	0,264	0,452	1,322	0,256	0,951	1,571	2,925	3,715
6_second	0,107	0,753	0,351	3,192	0,157	2,167	7,037	9,094	13,803
7_first	0,078	0,298	0,27	0,624	0,092	0,189	3,821	2,311	2,054
7_second	0,031	0,274	0,166	0,831	0,042	0,4	8,839	5,006	9,524
8_first	0,119	0,319	0,477	1,089	0,224	0,634	2,681	2,283	2,83
8_second	0,076	0,754	0,295	2,266	0,159	1,877	9,921	7,681	11,805
9_first	0,049	0,334	0,275	1,602	0,112	1,508	6,816	5,825	13,464
9_second	0,035	0,223	0,145	0,475	0,051	0,191	6,371	3,276	3,745
10_first	0,067	0,341	0,382	1,272	0,145	0,979	5,09	3,33	6,752

10_second	0,056	0,316	0,324	0,988	0,116	0,704	5,643	3,049	6,069
11_first	0,103	0,347	0,347	1,995	0,174	1,246	3,369	5,749	7,161
11_second	0,049	0,288	0,247	2,839	0,107	2,704	5,878	11,494	25,271
12_first	0,069	0,458	0,441	4,081	0,217	3,102	6,638	9,254	14,295
12_second	0,115	0,528	0,581	3,559	0,328	2,765	4,591	6,126	8,43
13_first	0,097	0,464	0,305	2,144	0,157	1,802	4,784	7,03	11,478
13_second	0,096	0,354	0,438	2,379	0,18	1,695	3,688	5,432	9,417
14_first	0,086	0,478	0,192	1,801	0,116	1,337	5,558	9,38	11,526
14_second	0,09	0,477	0,368	1,461	0,284	1,007	5,3	3,97	3,546
15_first	0,145	0,409	0,566	1,994	0,332	1,31	2,821	3,523	3,946
15_second	0,123	0,365	0,419	1,722	0,213	1,15	2,967	4,11	5,399
16_first	0,154	0,299	0,545	1,655	0,301	1,032	1,942	3,037	3,429
16_second	0,101	0,391	0,363	1,608	0,164	1,304	3,871	4,43	7,951
17_first	0,03	0,217	0,212	0,829	0,041	0,289	7,233	3,91	7,049
17_second	0,058	0,208	0,286	0,653	0,109	0,143	3,586	2,283	1,312
18_first	0,05	0,148	0,279	0,475	0,123	0,255	2,96	1,703	2,073
18_second	0,142	0,288	0,5	0,97	0,257	0,434	2,028	1,94	1,689
19_first	0,144	0,257	0,372	0,592	0,22	0,348	1,785	1,591	1,582
19_second	0,069	0,229	0,283	0,691	0,102	0,383	3,319	2,442	3,755
20_first	0,102	0,223	0,422	0,95	0,254	0,692	2,186	2,251	2,724
20_second	0,124	0,324	0,478	1,143	0,25	0,51	2,613	2,391	2,04
Среднее	0,108	0,375	0,419	1,759	0,219	1,275	4,331	4,612	7,135

Как видно из результатов, карта и траектория, полученная методом RTAB-Mar с использованием восстановленных нейронной сетью карт глубин хуже, чем аналогичные карты и траектории, но полученные с использованием истинных глубин. Это справедливо для всех карт.

Для проверки разработанной функции сопоставления точек были использованы те же карт, но пол на этих картах был удален. Результаты, полученные на таких картах представлены в таблице 8.

Таблица 8. Результаты полунатурного исследования методов на коллекции из симулятора Habitat с удаленным полом.

Карта	RGBD AME	FCNN AME	RGBD AME (our)	FCNN AME (our)	RGBD ATE	FCNN ATE	CC: detla	AME: delta	ATE: delta
1_first	0,13	0,586	0,569	2,75	0,26	2,506	4,508	4,833	9,638
1_second	0,145	0,611	0,509	3,306	0,226	2,319	4,214	6,495	10,261
2_first	0,543	0,702	1,269	3,102	0,895	2,453	1,293	2,444	2,741
2_second	0,164	0,599	0,597	3,452	0,31	2,699	3,652	5,782	8,706
3_first	0,033	0,295	0,18	0,798	0,04	0,474	8,939	4,433	11,85
3_second	0,114	0,356	0,346	0,934	0,198	0,573	3,123	2,699	2,894
4_first	0,443	0,88	1,303	3,016	1,033	2,24	1,986	2,315	2,168
4_second	0,162	0,599	0,491	3,623	0,284	2,444	3,698	7,379	8,606
5_first	0,049	0,438	0,201	1,66	0,095	1,34	8,939	8,259	14,105
5_second	0,107	0,315	0,352	1,272	0,126	0,831	2,944	3,614	6,595
6_first	0,13	0,344	0,373	1,368	0,256	0,951	2,646	3,668	3,715
6_second	0,138	0,925	0,34	3,01	0,157	2,167	6,703	8,853	13,803
7_first	0,074	0,367	0,244	0,646	0,092	0,189	4,959	2,648	2,054
7_second	0,031	0,422	0,151	0,833	0,042	0,4	13,613	5,517	9,524
8_first	0,127	0,499	0,434	1,121	0,224	0,634	3,929	2,583	2,83
8_second	0,097	0,917	0,323	2,391	0,159	1,877	9,454	7,402	11,805
9_first	0,1	0,353	0,271	1,601	0,112	1,508	3,53	5,908	13,464
9_second	0,134	0,247	0,142	0,475	0,051	0,191	1,843	3,345	3,745
10_first	0,15	0,435	0,454	1,3	0,145	0,979	2,9	2,863	6,752
10_second	0,142	0,349	0,333	0,941	0,116	0,704	2,458	2,826	6,069
11_first	0,095	0,392	0,308	1,929	0,174	1,246	4,126	6,263	7,161

11_second	0,064	0,307	0,231	2,645	0,107	2,704	4,797	11,45	25,271
12_first	0,102	0,959	0,416	4,536	0,217	3,102	9,402	10,904	14,295
12_second	0,167	0,994	0,505	3,277	0,328	2,765	5,952	6,489	8,43
13_first	0,091	0,604	0,315	2,141	0,157	1,802	6,637	6,797	11,478
13_second	0,11	0,644	0,432	2,553	0,18	1,695	5,855	5,91	9,417
14_first	0,116	0,808	0,198	1,807	0,116	1,337	6,966	9,126	11,526
14_second	0,085	0,697	0,38	1,49	0,284	1,007	8,2	3,921	3,546
15_first	0,181	0,507	0,577	2,268	0,332	1,31	2,801	3,931	3,946
15_second	0,108	0,406	0,376	1,815	0,213	1,15	3,759	4,827	5,399
16_first	0,196	0,36	0,526	1,496	0,301	1,032	1,837	2,844	3,429
16_second	0,116	0,45	0,357	1,679	0,164	1,304	3,879	4,703	7,951
17_first	0,031	0,262	0,229	0,766	0,041	0,289	8,452	3,345	7,049
17_second	0,06	0,288	0,275	0,686	0,109	0,143	4,8	2,495	1,312
18_first	0,058	0,204	0,279	0,489	0,123	0,255	3,517	1,753	2,073
18_second	0,154	0,361	0,472	0,976	0,257	0,434	2,344	2,068	1,689
19_first	0,181	0,442	0,37	0,702	0,22	0,348	2,442	1,897	1,582
19_second	0,079	0,319	0,293	0,782	0,102	0,383	4,038	2,669	3,755
20_first	0,125	0,345	0,428	0,909	0,254	0,692	2,76	2,124	2,724
20_second	0,163	0,486	0,521	1,258	0,25	0,51	2,982	2,415	2,04
Среднее	0,132	0,502	0,409	1,795	0,219	1,275	4,772	4,745	7,135

Если сравнивать результаты таблицы 8 и таблицы 9 (оригинальных карт и с удаленным полом), то можно заметить то, стандартная метрика AME для оригинальных карт и соответствующих им карт с удаленным полом различается в несколько раз, в то время как разработанная метрика отличается незначительно. Такое поведение больше соответствует контексту задачи mvSLAM, поскольку ожидается, что метрика будет инвариантна к подобного рода модификациям карты

подобно метрикам, которые предназначены для оценки качества траекторий. Визуализация полученных карт представлена на Рис. 34-Рис. 35.

Оригинальные карты

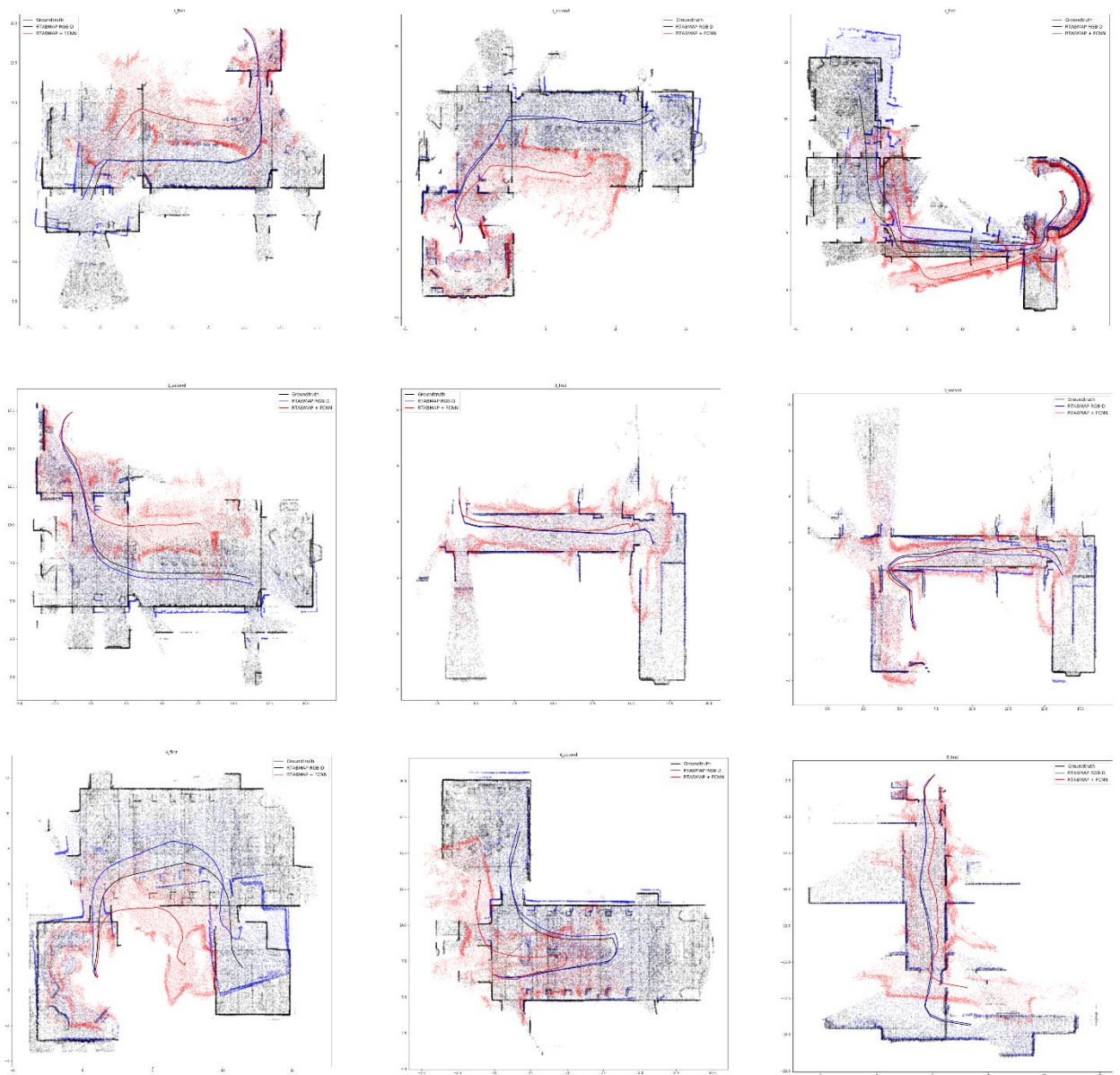


Рис. 34. Оригинальные карты и траектории. Черным цветом выделены истинные карты и траектории, синим – полученные RTAB-Map с использованием истинных глубин, красным – RTAB-Map с использованием FCNN.

Карты с удаленным полом

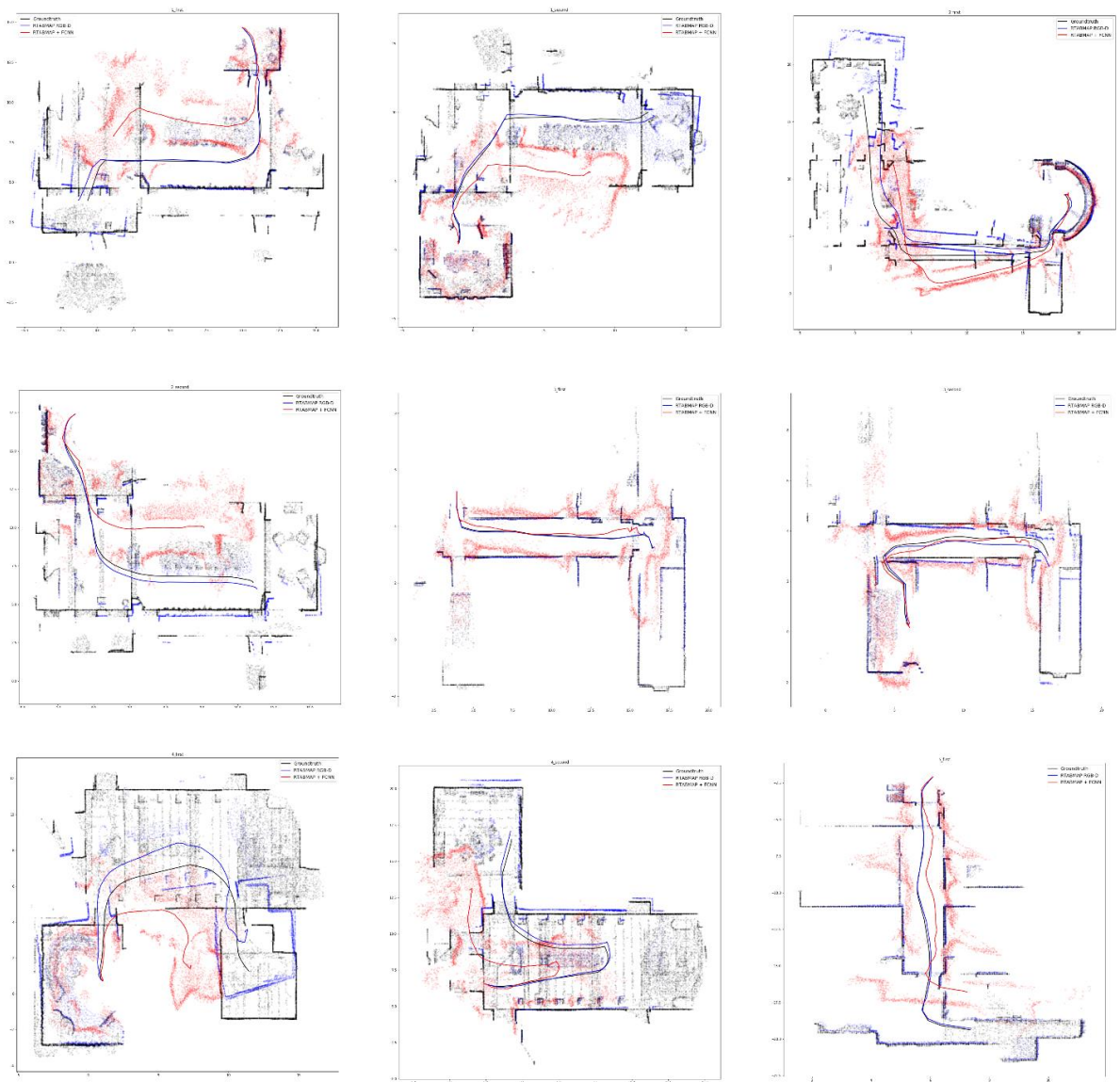


Рис. 35. Карты с удаленным полом.

Черным цветом выделены истинные карты и траектории, синим – полученные RTAB-Map с использованием истинных глубин, красным – RTAB-Map с использованием FCNN.

Для проверки полученного метода расчета сопоставлений был исследован алгоритм RTAB-Map. В качестве экспериментальной среды был выбран симулятор Habitat. На вход алгоритму подавались изображения и истинные карты глубин (RTAB-Map + Habitat depth), а также карты глубин, предсказанные архитектурой

нейронной сети, описанной в главе 2 (RTAB-Map + FCNN). Результаты расчета метрик отклонения траекторий ATE, RPE, а также среднеквадратичное отклонение с методом ближайшего соседа (AME), среднеквадратичное (AME (our)) и относительное (RME (our)) отклонения с разработанным методом сопоставления, представлены в таблице 6.

Таблица 9. Результаты оценки качества картирования и локализации.

Название	ATE	RPE	AME	AME (our)	RME (our)
RTAB-Map + Habitat depth	0.165	0.038	0.041	0.722	0.693
RTAB-Map + FCNN	0.667	0.068	0.361	1.935	1.891

В ходе эксперимента качество алгоритма RTAB-Map + FCNN получилось хуже по всем метрикам, чем качество алгоритма RTAB-Map с картами глубин из Habitat. При этом ошибка картирования по метрике CloudCompare у алгоритма RTAB-Map + FCNN в 9 раз больше, чем у RTAB-Map с известными картами глубин, а по метрикам AME (our) и RME (our) — примерно в 2.7 раза больше. То есть разница в качестве построенных карт в контексте задачи vSLAM не настолько существенна, как при прямом сравнении облаков точек. Это же подтверждает разница в ошибках локализации в построенной карте — отношение между абсолютными ошибками траектории двух алгоритмов примерно равно 4, между относительными ошибками позиции — примерно 1.8.

4.4 Выводы по главе

Основной проблемой при оценке качества картирования методов, решающих задачу картирования и локализации, является сопоставление точек в двух картах —

истинной и результирующей. Современные подходы, как правило, ищут соответствие на основе близости. Однако, такой подход работает нестабильно при ошибках масштаба, а также при типовых ошибках картирования. Таким образом, часто данные, полученные в результате подобного сравнения карт не репрезентативны. Для решения этой проблемы, в ходе диссертационной работы была разработана функция сопоставления трехмерных точек на основе истории позиций камеры, которая учитывает особенности задачи mvSLAM. Была разработана коллекция данных, состоящая из более 40 траектории, карт, а также наборов из изображений и карт глубин, полученных в фотореалистичном симуляторе Habitat, которые были использованы для проверки разработанной функции соответствия, а также для оценки качества работы алгоритма RTAB-Map с разными входными данными. Используются метрики оценки качества построения траектории, стандартные метрики сопоставления карт, а также разработанная функция сопоставления точек в составе стандартных метрик.

Полученные данные показали, что разработанная функция сопоставления позволяет вычислять метрики оценки качества картирования лучше, нежели использование стандартных функций (например, метод поиска ближайшего соседа или ICP). Рассмотренные базовые сценарии показали устойчивость полученной функции к удалению нерелевантных точек из финальной карты.

Заключение

Задача одновременного картирования и локализации по видеопотоку, полученному с единственной камеры, является важной задачей в областях компьютерного зрения и робототехники. Ее решение позволяет строить трехмерные карты неизвестной местности и восстанавливать позиции камеры на полученной карте, что в свою очередь позволяет повысить степень автономности робототехнических устройств. На основе перечисленных фактов было принято решение о теме диссертационного исследования.

В процессе выполнения работы было проведено исследование современных методов одновременного картирования и локализации по видеопотоку, полученному с единственной камеры. Выявлены основные преимущества и недостатки современных методов и алгоритмов, среди которых можно выделить: плохую детализацию карт, малое быстродействие, сложность оценки картирования.

Для решения проблемы низкого быстродействия, была исследована возможность ускорения современных алгоритмов поиска замыкания траекторий, входящих в состав методов mvSLAM,. Было предложено улучшение, позволяющее ограничить область поиска замыканий на основе истории позиций камеры. Предложенное улучшение было реализовано для алгоритмов LSD-SLAM и ORB-SLAM2 и экспериментально исследовано на открытых коллекциях данных. Эксперимент показал, что разработанный алгоритм позволил улучшить качество построения траектории на 8,42% и производительность (скорость работы) на 43,53% по сравнению с оригинальными алгоритмами, используемыми в методах для картирования и локализации по видеопотоку.

Для повышения детализированности карт, было предложено предсказывать карты глубин изображений с помощью искусственных нейронных сетей с последующим использованием их в методах, решающих задачу mvSLAM. Были предложены улучшения стандартных архитектур в частях кодировщика

(облегченная архитектура ResNet50) и декодировщика (использование non-bt сверток, ускоренная операция совмещения матриц), а также разработаны функции потерь для повышения качества обучения. Предложенные архитектуры были обучены на открытой коллекции данных NYU Dataset v2. Среди исследованных архитектур выбрана наиболее производительная по соотношению скорости работы к качеству восстановления глубин и реализована для встраиваемой платформы Nvidia Jetson TX2 в составе колесной робототехнической системы.

Для оценки качества картирования было предложено использование новой функции сопоставления двух множеств трехмерных точек (карт) на основе позиций камеры, которая в отличие от стандартных функций инвариантна к масштабу карты и типовым ошибкам картирования методов mvSLAM. Была разработана коллекция данных на основе фотореалистичного симулятора Habitat. Полученная функция сопоставления была протестирована в разработанной коллекции с алгоритмом RTAB-Map, на вход которому подавались истинные (полученные из Habitat) и восстановленные с помощью искусственных нейронных сетей карты глубин. Полученные данные показали, что разработанная функция сопоставления позволяет вычислять метрики оценки качества картирования лучше, нежели использование стандартных функций (например, метод поиска ближайшего соседа или ICP). Рассмотренные базовые сценарии показали устойчивость полученной функции к удалению нерелевантных точек из финальной карты.

Основные результаты:

- 1) Разработан новый алгоритм поиска замыканий траектории. Полученный алгоритм был исследован на открытых коллекциях данных с использованием реализаций методов LSD-SLAM и ORB-SLAM2. Сравнение проводилось с известными алгоритмами поиска замыканий по таким параметрам, как скорость работы, точность картирования и точность локализации. Исследование показало, что удалось добиться повышения точности локализации и построения карты на 8% и 7%

соответственно при уменьшении времени поиска замыкания траектории на 33%.

- 2) Разработана оригинальная модель искусственной нейронной сети для восстановления глубины изображений, которая применима для картирования и локализации в реальном времени на встраиваемой платформе Nvidia Jetson TX2. Проведено исследование разработанной модели на открытых коллекциях данных, которое показало увеличение быстродействия обработки одного кадра при сравнимой с современными моделями точностью восстановления глубины. Разработан программный комплекс картирования и локализации по видеопотоку в реальном времени, который был протестирован на встраиваемой платформе Nvidia Jetson TX2 и на реальной робототехнической системе.
- 3) Предложена новая модель оценки качества карт, полученных методами, решающие задачу mvSLAM, на основе оригинальной функции сопоставления трехмерных точек с учетом истории позиций камеры. Применение предложенной модели позволяет повысить точность сравнения карт, представимых в виде облаков точек.

В качестве основных направлений дальнейших исследований можно выделить следующие:

- 1) Повышения качества картирования и локализации методов, решающих задачу mvSLAM, с использованием искусственных нейронных сетей. Возможно рассмотрение архитектур позволяющих восстанавливать не только карты глубин изображений, но и позиций камеры;
- 2) Расширение экспериментальной базы для оценки качества методов картирования и локализации по видеопотоку с использованием разработанных метрик;
- 3) Программная оптимизация реализации методов, решающих задачу mvSLAM, для увеличения быстродействия на встраиваемых системах и бортовых вычислителях;

- 4) Разработка методов визуализации данных, полученных методами, решающие задачу mvSLAM;
- 5) Применение разработанных методов в составе системы автономной навигации сложных робототехнических устройств.

Список публикаций автора

1. Боковой А.В. Архитектура системы автоматического управления группой малых беспилотных летательных аппаратов // Информационные технологии и вычислительные системы. – 2018. – №. 1. – С. 68-77.
2. Боковой А.В., Муравьев К.Ф., Яковлев К.С. Система одновременного картирования, локализации и исследования неизвестной местности по видеопотоку // Информационные технологии и вычислительные системы. – 2020. – №. 2. – С. 51-61.
3. Bokovoy A., Yakovlev K. Enhancing semi-dense monocular vSLAM used for multi-rotor UAV navigation in indoor environment by fusing IMU data // The 2018 International Conference on Artificial Life and Robotics (ICAROB2018). 2018. P. 391-394.
4. Bokovoy A., Yakovlev K. Original Loop-closure Detection Algorithm for Monocular vSLAM // 6th International Conference Analysis of Images, Social Networks and Texts (AIST 2017). 2017. P. 202-213.
5. Bokovoy A., Fomin B., Yakovlev K. Implementation of the Pathfinding System for Autonomous Navigation of Mobile Ground Robot Selected // Papers of the 1st Workshop (Summer Session) at the Conference on Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems (ITTMM-WSS-2018). 2018. P.72-78.
6. Bokovoy A., Yakovlev K. Sparse 3D Point-Cloud Map Upsampling and Noise Removal as a vSLAM Post-Processing Step: Experimental Evaluation // International Conference on Interactive Collaborative Robotics. – Springer, Cham. 2018. P. 23-33.
7. Bokovoy A., Muravyev K., Yakovlev K. Real-time Vision-based Depth Reconstruction with NVidia Jetson // 2019 European Conference on Mobile Robots (ECMR). – IEEE. 2019. P. 1-6.
8. Bokovoy A., Muraviev K., Yakovlev K. Map-merging algorithms for visual SLAM: feasibility study and empirical evaluation // Russian Conference on Artificial Intelligence. – Springer, Cham, 2020. – P. 46-60.
9. Muravyev K., Bokovoy A., Yakovlev K. Enhancing exploration algorithms for navigation with visual SLAM // Russian Conference on Artificial Intelligence. – Springer, Cham, 2021. – P. 197-212.

10. Bokovoy A., Muraviev K. Assessment of Map Construction in vSLAM // 2021 International Siberian Conference on Control and Communications (SIBCON). – IEEE, 2021. – P. 1-6.
11. Muravyev K., Bokovoy A., Yakovlev K. tx2_fcnn_node: An open-source ROS compatible tool for monocular depth reconstruction // SoftwareX. – 2022. – P. 1-6.
12. Боковой А.В. Распознавание маркеров на изображениях, полученных с нескольких малых беспилотных летательных аппаратов // XV национальная конференция по искусственному интеллекту с международным участием КИИ-2016 (3–7 октября 2016 г., г. Смоленск, Россия). Труды конференции. В 3-х томах., Смоленск: Универсум, 2016, 3, С. 245-254.
13. Боковой А.В. Совмещение границ в методах одновременного картирования и локализации по видеопотоку, полученному с единственной камеры // VI Всероссийская конференция (с международным участием) «Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем» (ИТТиММ-2016), Москва, 18-20 апреля 2016, С. 133-135.
14. Боковой А.В. Методы и алгоритмы визуальной навигации группы беспилотных летательных аппаратов по множеству маркеров // Сборник научных трудов IV Международной летней школы-семинара по искусственному интеллекту для студентов, аспирантов, молодых ученых и специалистов (Санкт-Петербург, 30 июня – 3 июля 2017 г.). — СПб.: Политехника-сервис, 2017. — С. 32-40.
15. Боковой А.В. Метод решения задачи одновременного картирования и локализации по видеопотоку и данным инерциальной навигационной системы для малых БПЛА // Четвертый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2017, 5-6 октября 2017 г., г. Казань, Республика Татарстан, Россия): тр. семинара / под ред. Е.А. Магида, В.Е. Павловского, К.С. Яковлева. – Казань: Центр инновационных технологий, 2017. – С. 36-45.
16. Муравьев К.Ф., Боковой А.В. Восстановление карт глубин изображений, полученных с единственной видеокамеры в реальном времени на платформе NVIDIA JETSON TX2 // Пятый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного

интеллекта» (БТС-ИИ-2019): Труды семинара. – Переславль-Залесский: Российская ассоциация искусственного интеллекта, 2019. С. 29-39.

17. Муравьев К.Ф., Боковой А.В., Яковлев К.С. Оценка качества алгоритмов картирования и локализации на основе видеоданных в симуляционных средах // Завалишинские чтения'20: XV Международная конференция по электромеханике и робототехнике (СПб., 15-18 апреля 2020 г.): сб. докл. - СПб.: ГУАП, 2020. С. 129-136.

Список литературы

1. Abdellali H., Kato Z. Absolute and relative pose estimation of a multi-view camera system using 2d-3d line pairs and vertical direction //2018 Digital Image Computing: Techniques and Applications (DICTA). – IEEE, 2018. – С. 1-8.
2. Angeli A. et al. Real-time visual loop-closure detection //2008 IEEE international conference on robotics and automation. – IEEE, 2008. – С. 1842-1847.
3. Badino H., Huber D., Kanade T. Visual topometric localization //2011 IEEE Intelligent Vehicles Symposium (IV). – IEEE, 2011. – С. 794-799.
4. Bay H., Tuytelaars, T., Van Gool, L., SURF: Speeded Up Robust Features. //Proceedings of the ninth European Conference on Computer Vision, 2006.
5. Blanco-Claraco J. L., Moreno-Duenas F. A., González-Jiménez J. The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario //The International Journal of Robotics Research. – 2014. – Т. 33. – №. 2. – С. 207-214.
6. Bloesch M. et al. CodeSLAM—learning a compact, optimisable representation for dense visual SLAM //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2018. – С. 2560-2568.
7. Bortoff S. A. Path planning for UAVs //Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334). – IEEE, 2000. – Т. 1. – №. 6. – С. 364-368.
8. Bouaziz S., Tagliasacchi A., Pauly M. Sparse iterative closest point //Computer graphics forum. – Oxford, UK : Blackwell Publishing Ltd, 2013. – Т. 32. – №. 5. – С. 113-123.
9. Bradski G., Kaehler A. OpenCV //Dr. Dobb's journal of software tools. – 2000. – Т. 3.
10. Burguera A., Oliver G., González Y. Scan-based SLAM with trajectory correction in underwater environments //2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2010. – С. 2546-2551.

11. Burri M. et al. The EuRoC micro aerial vehicle datasets //The International Journal of Robotics Research. – 2016. – T. 35. – №. 10. – C. 1157-1163.
12. Campoy P. et al. Computer vision onboard UAVs for civilian tasks //Journal of Intelligent and Robotic Systems. – 2009. – T. 54. – №. 1. – C. 105-135.
13. Cesare K. et al. Multi-UAV exploration with limited communication and battery //2015 IEEE international conference on robotics and automation (ICRA). – IEEE, 2015. – C. 2230-2235.
14. Chang A. Matterport3d: Learning from rgb-d data in indoor environments //arXiv preprint arXiv:1709.06158. 2017.
15. Chen Y. et al. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks //IEEE Transactions on Geoscience and Remote Sensing. – 2016. – T. 54. – №. 10. – C. 6232-6251.
16. Civera J., Davison A. J., Montiel J. M. M. Inverse depth parametrization for monocular SLAM //IEEE transactions on robotics. – 2008. – T. 24. – №. 5. – C. 932-945.
17. Civera J., Davison A. J., Montiel J. M. M. Inverse depth to depth conversion for monocular slam //Proceedings 2007 IEEE International Conference on Robotics and Automation. – IEEE, 2007. – C. 2778-2783.
18. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2017. – T. 2
19. Cui H., Dahnoun N. Real-Time Stereo Vision Implementation on Nvidia Jetson TX2 //2019 8th Mediterranean Conference on Embedded Computing (MECO). – IEEE, 2019. – C. 1-5.
20. Datta, J.S. Kim, T. Kanade. Accurate camera calibration using iterative refinement of control points.// Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on 2009, p. 1201-1208
21. Davison A, Reid I, Molton N. and Stasse O. MonoSLAM: Real-Time Single Camera SLAM // IEEE Transactions on Pattern Analysis and Machine Intelligence archive, Volume 29 Issue 6, J, pp1052-1067, 2007

22. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 1119-1127.
23. Dong J. et al. VL-SLAM: Real-Time Visual-Inertial Navigation and Semantic Mapping //IEEE Conference on Computer Vision and Pattern Recognition, Live Demo. – 2016.
24. Duan C. et al. Deep learning for visual SLAM in transportation robotics: a review //Transportation Safety and Environment. – 2019. – T. 1. – №. 3. – C. 177-184.
25. Engel J., Koltun V., Cremers D. Direct sparse odometry //IEEE transactions on pattern analysis and machine intelligence. – 2017. – T. 40. – №. 3. – C. 611-625.
26. Engel J., Schops T. and Cremers D. LSD-LSAM: Large-Scale Monocular SLAM // In Proc. European Conference on Computer Vision (ECCV), 2007
27. Forster C. et al. SVO: Semidirect visual odometry for monocular and multicamera systems //IEEE Transactions on Robotics. – 2016. – T. 33. – №. 2. – C. 249-265.
28. Gallardo N. et al. Formation control implementation using kobuki turtlebots and parrot bebop drone //2016 World Automation Congress (WAC). – IEEE, 2016. – C. 1-6.
29. Garg R. et al. Unsupervised cnn for single view depth estimation: Geometry to the rescue //European Conference on Computer Vision. – Springer, Cham, 2016. – C. 740-756.
30. Garg, R., BG, V. K., Carneiro, G., & Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue //European Conference on Computer Vision. – Springer, Cham, 2016. – C. 740-756.
31. Gebrehiwot A. A., Hashemi-Beni L. Three-Dimensional Inundation Mapping Using UAV Image Segmentation and Digital Surface Model //ISPRS International Journal of Geo-Information. – 2021. – T. 10. – №. 3. – C. 144.
32. Geiger A. et al. Vision meets robotics: The kitti dataset //The International Journal of Robotics Research. – 2013. – T. 32. – №. 11. – C. 1231-1237.
33. Giubilato R. et al. An evaluation of ros-compatible stereo visual slam methods on a nvidia jetson tx2 //Measurement. – 2019. – T. 140. – C. 161-170.

34. Greene W. N. et al. Multi-level mapping: Real-time dense monocular SLAM //2016 IEEE International Conference on Robotics and Automation (ICRA). – IEEE, 2016. – C. 833-840.
35. Grisetti G. et al. g2o: A general framework for (hyper) graph optimization //Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China. – 2011. – C. 9-13.
36. Handa A. et al. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM //2014 IEEE international conference on Robotics and automation (ICRA). – IEEE, 2014. – C. 1524-1531.
37. Hartley I, In Defence of the 8-point Algorithm. // Pattern Analysis and Machine Intelligence, Vol. 19, Issue 6, pp. 580-593, 1997
38. Hazirbas C., Leal-Taixé L., Cremers D. Deep Depth From Focus // arXiv preprint arXiv:1704.01085. – 2017.
39. He K., Sun J., Tang X. Guided image filtering //European conference on computer vision. – Springer, Berlin, Heidelberg, 2010. – C. 1-14.
40. Ho K. L., Newman P. Detecting loop closure with scene sequences //International journal of computer vision. – 2007. – T. 74. – №. 3. – C. 261-286.
41. Ho K. L., Newman P. Loop closure detection in SLAM by combining visual and spatial appearance //Robotics and Autonomous Systems. – 2006. – T. 54. – №. 9. – C. 740-749.
42. Honegger D., Sattler T., Pollefeys M. Embedded real-time multi-baseline stereo //2017 IEEE International Conference on Robotics and Automation (ICRA). – IEEE, 2017. – C. 5245-5250.
43. Hwang S. Y., Song J. B. Monocular vision-based SLAM in indoor environment using corner, lamp, and door features from upward-looking camera //IEEE Transactions on Industrial Electronics. – 2011. – T. 58. – №. 10. – C. 4804-4812.
44. Jin X. et al. Estimates of plant density of wheat crops at emergence from very low altitude UAV imagery //Remote Sensing of Environment. – 2017. – T. 198. – C. 105-114.

45. Jin, Y., Yu, L., Chen, Z., & Fei, S. (2021). A Mono SLAM Method Based on Depth Estimation by DenseNet-CNN// IEEE Sensors Journal.
46. Julier, S.J.; Uhlmann, J.K.. Unscented filtering and nonlinear estimation.//Proceedings of the IEEE, pp. 401–422, 2004
47. Karamchandani S. et al. SLAM Using Neural Network-Based Depth Estimation for Auto Vehicle Parking //IOT with Smart Systems. – Springer, Singapore, 2022. – C. 37-44.
48. Kerl C., Sturm J., Cremers D. Dense visual SLAM for RGB-D cameras // Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. – IEEE, 2013. – C. 2100-2106.
49. Kim B. D. et al. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network //2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). – IEEE, 2017. – C. 399-404.
50. Klein G. and Murray D. Parallel Tracking and Mapping for Small AR Workspaces // In Proc. International Symposium on Mixed and Augmented Reality, ISMAR'07, 2007
51. Koenderink J. J., Van Doorn A. J. Affine structure from motion //JOSA A. – 1991. – T. 8. – №. 2. – C. 377-385.
52. Koenig N., Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator //2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). – IEEE, 2004. – T. 3. – C. 2149-2154.
53. Kovesi P. et al. Image features from phase congruency //Videre: Journal of computer vision research. – 1999. – T. 1. – №. 3. – C. 1-26.
54. Labbé M., Michaud F. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation //Journal of Field Robotics. – 2019. – T. 36. – №. 2. – C. 416-446.
55. Laidlow T., Czarnowski J., Leutenegger S. DeepFusion: Real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient

- predictions //2019 International Conference on Robotics and Automation (ICRA). – IEEE, 2019. – C. 4068-4074.
56. Laina I. et al. Deeper depth prediction with fully convolutional residual networks //2016 Fourth international conference on 3D vision (3DV). – IEEE, 2016. – C. 239-248.
 57. Lemaire T., Berger C., Jung I. and Lacroix S. Vision-Based SLAM: Stereo and Monocular Approaches // International Journal of Computer Vision, Volume 74, Issue 3, pp 343-364, 2007
 58. Li et al., Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 1119-1127.
 59. Li J. et al. NRLI-UAV: Non-rigid registration of sequential raw laser scans and images for low-cost UAV LiDAR point cloud quality improvement //ISPRS Journal of Photogrammetry and Remote Sensing. – 2019. – T. 158. – C. 123-145.
 60. Li Q. et al. Robust scale-invariant feature matching for remote sensing image registration //IEEE Geoscience and Remote Sensing Letters. – 2009. – T. 6. – №. 2. – C. 287-291.
 61. Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 3431-3440.
 62. Lowe G., Distinctive Image Features from Scale-Invariant Keypoints. // International Journal of Computer Vision, pp. 91-110, 2004
 63. Lu Y. et al. A survey on vision-based UAV navigation //Geo-spatial information science. – 2018. – T. 21. – №. 1. – C. 21-32.
 64. Matsuki H. et al. CodeMapping: Real-Time Dense Mapping for Sparse SLAM using Compact Scene Representations //IEEE Robotics and Automation Letters. – 2021. – T. 6. – №. 4. – C. 7105-7112.
 65. Meyer J. et al. Comprehensive simulation of quadrotor uavs using ros and gazebo //International conference on simulation, modeling, and programming for autonomous robots. – Springer, Berlin, Heidelberg, 2012. – C. 400-411.

66. Morel J. M., Yu G. ASIFT: A new framework for fully affine invariant image comparison //SIAM journal on imaging sciences. – 2009. – T. 2. – №. 2. – C. 438-469.
67. Munguía R. et al. Vision-based SLAM system for unmanned aerial vehicles //Sensors. – 2016. – T. 16. – №. 3. – C. 372.
68. Mur-Artal R., Tardós J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras // IEEE Transactions on Robotics. – 2017. – T. 33. – №. 5. – C. 1255-1262.
69. Neira J, Tardos J. Data Association in Stochastic Mapping using the Joint Compatibility Test. // Robot and Automation, IEEE Transactions pp. 890-897, 2001
70. Newcombe R. A., Lovegrove S. J., Davison A. J. DTAM: Dense tracking and mapping in real-time //2011 international conference on computer vision. – IEEE, 2011. – C. 2320-2327.
71. Pan X., Lyu S. Region duplication detection using image feature matching //IEEE Transactions on Information Forensics and Security. – 2010. – T. 5. – №. 4. – C. 857-867.
72. Pannu G. S., Ansari M. D., Gupta P. Design and implementation of autonomous car using Raspberry Pi //International Journal of Computer Applications. – 2015. – T. 113. – №. 9.
73. Parida K. K., Srivastava S., Sharma G. Beyond image to depth: Improving depth prediction using echoes //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2021. – C. 8268-8277.
74. Qin T., Li P., Shen S. Relocalization, global optimization and map merging for monocular visual-inertial SLAM //2018 IEEE International Conference on Robotics and Automation (ICRA). – IEEE, 2018. – C. 1197-1204.
75. Quigley M. et al. ROS: an open-source Robot Operating System //ICRA workshop on open source software. – 2009. – T. 3. – №. 3.2. – C. 5.

76. Romera E. et al. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation //IEEE Transactions on Intelligent Transportation Systems. – 2017. – T. 19. – №. 1. – C. 263-272.
77. Rosten E. and Drummond T., Machine learning for high-speed corner detection. //In Proc. 9th European Conference on Computer Vision (ECCV'06), 2006.
78. Rusu R. B., Cousins S. 3d is here: Point cloud library (pcl) //2011 IEEE international conference on robotics and automation. – IEEE, 2011. – C. 1-4.
79. Savva M. et al. Habitat: A platform for embodied ai research //Proceedings of the IEEE/CVF International Conference on Computer Vision. – 2019. – C. 9339-9347.
80. Scherer S. A., Zell A. Efficient onboard RGBD-SLAM for autonomous MAVs //2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2013. – C. 1062-1068.
81. Schönberger J. L. et al. Semantic visual localization //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – C. 6896-6906.
82. Schops T., Sattler T., Pollefeys M. Bad slam: Bundle adjusted direct rgb-d slam //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – C. 134-144.
83. Schops T., Sattler T., Pollefeys M. Bad slam: Bundle adjusted direct rgb-d slam //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – C. 134-144.
84. Silveira G., Malis E., Rives P. An efficient direct approach to visual SLAM //IEEE transactions on robotics. – 2008. – T. 24. – №. 5. – C. 969-979.
85. Song S., Lichtenberg S. P., Xiao J. Sun rgb-d: A rgb-d scene understanding benchmark suite //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 567-576.
86. Song Z., Woodruff D. P., Zhong P. Relative error tensor low rank approximation //Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. – Society for Industrial and Applied Mathematics, 2019. – C. 2772-2789.

87. Strasdat H., Montiel J. M. M., Davison A. J. Real-time monocular SLAM: Why filter? //2010 IEEE International Conference on Robotics and Automation. – IEEE, 2010. – C. 2657-2664.
88. Strasdat H., Montiel J., Davison A. J. Scale drift-aware large scale monocular SLAM //Robotics: Science and Systems VI. – 2010. – T. 2. – №. 3. – C. 7.
89. Trujillo J. C. et al. Cooperative monocular-based SLAM for multi-UAV systems in GPS-denied environments //Sensors. – 2018. – T. 18. – №. 5. – C. 1351.
90. Ullman S. The interpretation of structure from motion //Proceedings of the Royal Society of London. Series B. Biological Sciences. – 1979. – T. 203. – №. 1153. – C. 405-426.
91. Van De Ville D. et al. Noise reduction by fuzzy image filtering //IEEE transactions on fuzzy systems. – 2003. – T. 11. – №. 4. – C. 429-436.
92. Walter M. R., Eustice R. M., Leonard J. J. Exactly sparse extended information filters for feature-based SLAM //The International Journal of Robotics Research. – 2007. – T. 26. – №. 4. – C. 335-359.
93. Wang J., Lin J., Wang Z. Efficient convolution architectures for convolutional neural network //2016 8th International Conference on Wireless Communications & Signal Processing (WCSP). – IEEE, 2016. – C. 1-5.
94. Wang R. et al. License plate localization in complex scenes based on oriented FAST and rotated BRIEF feature //Journal of Electronic Imaging. – 2015. – T. 24. – №. 5. – C. 053011.
95. Wang S. et al. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks //2017 IEEE international conference on robotics and automation (ICRA). – IEEE, 2017. – C. 2043-2050.
96. Wang W., Lu Y. Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model //IOP conference series: materials science and engineering. – IOP Publishing, 2018. – T. 324. – №. 1. – C. 012049.

97. Whitehill J., Omlin C. W. Haar features for FACS AU recognition //7th international conference on automatic face and gesture recognition (FGR06). – IEEE, 2006. – C. 5 pp.-101.
98. Wolcott R. W., Eustice R. M. Visual localization within lidar maps for automated urban driving //2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2014. – C. 176-183.
99. Wu Z., Shen C., Van Den Hengel A. Wider or deeper: Revisiting the resnet model for visual recognition //Pattern Recognition. – 2019. – T. 90. – C. 119-133.
100. Wurm K. M. et al. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems //Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation. – 2010. – T. 2.
101. Xia F. Gibson env: Real-world perception for embodied agents // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. P. 9068-9079.
102. Xu Y. et al. 3D point cloud map based vehicle localization using stereo camera //2017 IEEE Intelligent Vehicles Symposium (IV). – IEEE, 2017. – C. 487-492.
103. Ye X. et al. DRM-SLAM: Towards dense reconstruction of monocular SLAM with scene depth fusion //Neurocomputing. – 2020. – T. 396. – C. 76-91.
104. Yin W. et al. Enforcing geometric constraints of virtual normal for depth prediction //Proceedings of the IEEE/CVF International Conference on Computer Vision. – 2019. – C. 5684-5693.
105. Zhang Y. et al. Image super-resolution using very deep residual channel attention networks //Proceedings of the European conference on computer vision (ECCV). – 2018. – C. 286-301.
106. Zhang Z. A flexible new technique for camera calibration //IEEE Transactions on pattern analysis and machine intelligence. – 2000. – T. 22. – №. 11. – C. 1330-1334.
107. Zhang Z., Scaramuzza D. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry //2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – IEEE, 2018. – C. 7244-7251.

108. Zhao Y., Vela P. A. Good feature selection for least squares pose optimization in VO/VSLAM //2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – IEEE, 2018. – С. 1183-1189.
109. Андрейчук А. А., Яковлев К. С. Планирование траектории на плоскости с учетом размера агента (мобильного робота, беспилотного транспортного средства) // Четвертый Всероссийский научно-практический семинар "Беспилотные транспортные средства с элементами искусственного интеллекта" (БТС-ИИ-2017). – 2017. – С. 107-117.
110. Федеральный исследовательский центр Информатика и управление РАН [Электронный ресурс]: сайт. – Москва: ФИЦ ИУ РАН. – URL: <http://hhprcc.frccsc.ru> (дата обращения: 12.09.2018)

Список рисунков

Рис. 1. Пример карты, построенной непрямым методом ORB-SLAM2.....	15
Рис. 2. Примеры выделения особенностей с помощью методов SIFT, SURF и FAST-10	16
Рис. 3. Карта, построенная с использованием контрастных областей изображений (LSD-SLAM Foodcourt Dataset). Изображение из [26].	17
Рис. 4. Карта глубины, составленная по паре изображений (Middlebury Stereo Datasets). 1 – изображение с левой камеры, 2 – изображение с правой камеры, 3, 4 – соответствующие им карты глубины. Чем темнее пиксель на карте глубины – тем дальше объект.....	18
Рис. 5. Примеры карт, построенных с помощью методов MonoSLAM, PTAM и LSD-SLAM	20
Рис. 6. Пример работы прямых методов. Изображение из [25].....	22
Рис. 7. Исходное изображение (слева) и восстановленная карта глубины (справа). Яркость пикселей на изображении справа означают расстояния до соответствующих объектов (ярче - дальше).....	24
Рис. 8. Архитектура сети Deep Depth From Focus Net (DDFFNet).	26
Рис. 9. Примеры карт и траекторий, построенных с помощью метода LSD-SLAM. A- Foodcoutr Sequence, B- ECCV Sequence.....	32
Рис. 10. 3D модель местности из набора видео Living Room Dataset.....	33
Рис. 11. Экспериментальная установка.....	35
Рис. 12. Схематичная иллюстрация проблемы обнаружения замыканий на траектории (слева) и глобальной оптимизации (справа) с целью корректировки накопившийся ошибки. (Черная кривая – траектория, зеленые точки – позиции камеры, оранжевые точки – карта).....	42
Рис. 13. Иллюстрация области поиска замыканий траектории	45

Рис. 14. График относительной ошибки траектории для всех коллекций в %	53
Рис. 15. График ошибки поворота позиции в град./м. для всех траекторий	54
Рис. 16. Абсолютная ошибка траектории для всех коллекций в метрах	54
Рис. 17. График относительной ошибки траектории для всех коллекций в метрах.	55
Рис. 18. Среднее время работы алгоритма поиска замыканий на траектории в секундах для всех коллекций.	56
Рис. 19. Картинка (слева) и соответствующая ей карта глубины (справа). Значения каждого пикселя карты глубины является расстоянием до советующего объекта.	59
Рис. 20. Визуализация исследованных архитектур.	61
Рис. 21. Замена операции сверточного повышения дискретизации на чередование. (Слева – блок сверточного повышения дискретизации, который состоит из операций повышения частоты дискретизации, свертки с ядром 5x5 и активации. Справа – эквивалентная замена из серии сверток, операции объединения весов чередованием и активации).	65
Рис. 22. Иллюстрация замены свертки на операцию чередования	66
Рис. 23. Иллюстрация проблемы восстановления глубин на близких дистанциях. Красными прямоугольниками выделены области, на которых предсказанные глубины сильно отличаются от истинных.	68
Рис. 24. Качество карт глубин, восстановленных архитектурой Lite Basic + SC на изображениях коллекции NYU Dataset v2.	72
Рис. 25. Реализованная на встраиваемой платформе архитектура искусственной нейронной сети.	72
Рис. 26. Пример работы полученной системы на примере реальной робототехнической системы	74
Рис. 27. Пример модели помещения в симуляторе Gazebo	80

Рис. 28. Прохождение света сквозь стену в симуляторе Gazebo	81
Рис. 29. Примеры модели помещения из коллекции Matterport 3D в симуляторе Habitat	82
Рис. 30. Визуализация проблемы сравнения карт стандартными метриками. (А) Показана истинная среда (зеленые и синие точки) и углы обзора камеры в разные моменты времени; Б) Построенная карта методами vSLAM (зеленые и красные точки); В) Истинные соответствия между точками (синие стрелки) и вычисленные методом ближайшего соседа (оранжевые стрелки)).	84
Рис. 31. Истинные карты и траектории, а также результаты работы методов vSLAM. а) Оригинальные результаты; б) Результаты с удаленным полом.	86
Рис. 32. Пример расчета соответствия.	88
Рис. 33. vSLAM. Слева направо: истинная карта помещения (восстановлена методом, описанным в разделе 3), карта, построенная алгоритмом RTAB-Map по изображениям и глубинам из Habitat; карта, построенная алгоритмом RTAB-Map + FCNN.....	90
Рис. 34. Оригинальные карты и траектории. Черным цветом выделены истинные карты и траектории, синим – полученные RTAB-Map с использованием истинных глубин, красным – RTAB-Map с использованием FCNN.	95
Рис. 35. Карты с удаленным полом. Черным цветом выделены истинные карты и траектории, синим – полученные RTAB-Map с использованием истинных глубин, красным – RTAB-Map с использованием FCNN.....	96

Список таблиц

Таблица 1. Результаты экспериментального исследования не прямых методов vSLAM.	20
Таблица 2. Параметры коллекций RGB-D SLAM Dataset and Benchmark, LSD-SLAM Dataset, ICL-NUIM Dataset и Malaga Dataset 2009	36
Таблица 3. Таблица срабатываний метода поиска замыканий траекторий в зависимости от количества особых точек K	50
Таблица 4. Средние результаты тестирования разработанных алгоритмов по сравнению с оригинальными.....	56
Таблица 5. Улучшение показателей относительно оригинальных методов в %	57
Таблица 6. Результаты экспериментального исследования разработанных архитектур искусственных нейронных сетей.	71
Таблица 7. Результаты полунатурного исследования методов на оригинальной коллекции из симулятора Habitat	91
Таблица 8. Результаты полунатурного исследования методов на коллекции из симулятора Habitat с удаленным полом.....	93
Таблица 9. Результаты оценки качества картирования и локализации.....	97