

На правах рукописи



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»

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Москва – 2022

Работа выполнена на кафедре информационных технологий Российского университета дружбы народов.

Научный руководитель: **Яковлев Константин Сергеевич**
кандидат физико-математических наук

Официальные оппоненты: **Вохминцев Александр Владиславович**
доктор технических наук
Челябинский государственный университет,
профессор кафедры информационных технологий и экономической информатики

Степанов Дмитрий Николаевич
кандидат технических наук
Институт программных систем им. А.К. Айламазяна Российской академии наук,
научный сотрудник исследовательского центра мультипроцессорных систем

Ведущая организация: Институт проблем управления им. В.А. Трапезникова Российской академии наук

Защита состоится ____ 2022 г. в ____ часов ____ минут на заседании диссертационного совета 24.1.224.01 на базе Федерального государственного учреждения «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН) по адресу 117312, г. Москва, проспект 60-летия Октября, 9 (конференц-зал, 1-й этаж).

С диссертацией можно ознакомиться в библиотеке ФИЦ ИУ РАН по адресу: г. Москва, ул. Вавилова, д. 40 и на официальном сайте ФИЦ ИУ РАН <http://www.frccsc.ru>.

Отзывы на автореферат в двух экземплярах, заверенные печатью учреждения, выслать по адресу: 119333, г. Москва, ул. Вавилова, д. 44, кор. 2, ученому секретарю диссертационного совета 24.1.224.01.

Автореферат разослан ____ 2022 г.
Телефон для справок: +7 (499) 135-51-64

Ученый секретарь диссертационного совета
24.1.224.01,
кандидат физико-математических наук, доцент

И.В. Смирнов

Общая характеристика работы

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 относится 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.

Forester, M. Pizzoli, D. Scaramuzza, K. Tateno, F. Tombari, I. Laina, N. Navab, E.H. 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.

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 ____ страниц, включая ____ рисунков и ____ таблиц. Список литературы содержит ____ 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.

Содержание работы.

Во **введении** обосновывается актуальность диссертационной работы и ее научная новизна, формулируются основные цели и задачи диссертационного исследования, приводятся положения, выносимые на защиту, дается краткое содержание глав диссертации.

Первая глава описывает принципы работы современных методов, решающих задачу одновременного картирования и локализации по видеопотоку. В главе рассмотрены особенности их применения в задаче автономной навигации мобильной робототехники. Рассмотрено применение искусственных нейронных сетей для задачи восстановления глубины изображений в реальном времени на мобильных платформах с целью использования при решении задачи mvSLAM.

На основе проведенного исследования уточнены задачи диссертационной работы.

Вторая глава содержит постановку задачи обнаружения замыкания траекторий в контексте mvSLAM. Описан предлагаемый оригинальный алгоритм поиска замыкания траектории.

Алгоритмы замыкания траекторий состоят из двух процедур: 1) поиск замыканий, 2) глобальная оптимизация. Задачей поиска замыкания траектории является определение мест, в которых робототехническая система уже была в процессе картирования и локализации. После обнаружения замыкания выполняется глобальная оптимизация трехмерной карты и полученной траектории с целью устранения накопившейся ошибки. Упрощенный пример работы алгоритма замыкания траекторий изображен на рис. 1.

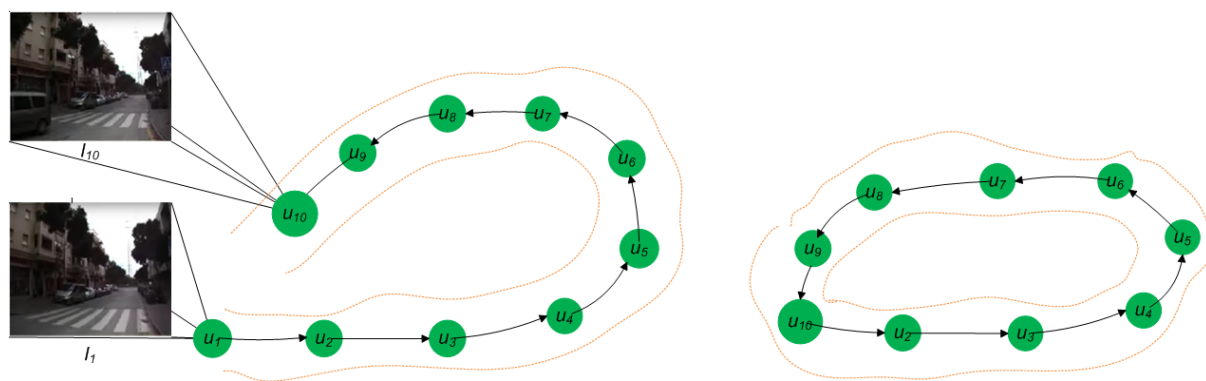


Рис. 1. Схематическая иллюстрация решения задачи обнаружения замыканий на траектории (слева) и глобальной оптимизации (справа) с целью корректировки накопившейся ошибки (черная кривая – траектория, зеленые точки – позиции камеры, оранжевые точки – карта).

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 накопившейся ошибки.

Как правило, известные алгоритмы обнаружения замыканий траектории сравнивают текущее изображение со всеми предыдущими на видеопотоке (рис. 2.А), что ведет к увеличивающейся со временем вычислительной сложности процедуры обнаружения замыканий и к менее точному сопоставлению изображений. Это может негативно сказываться на качестве (точности) результирующих карт и траекторий.

Для устранения указанных недостатков было предложено использовать информации о движении робототехнической системы для эвристического ограничения области поиска, к которой принадлежат изображения для последующего сопоставления.

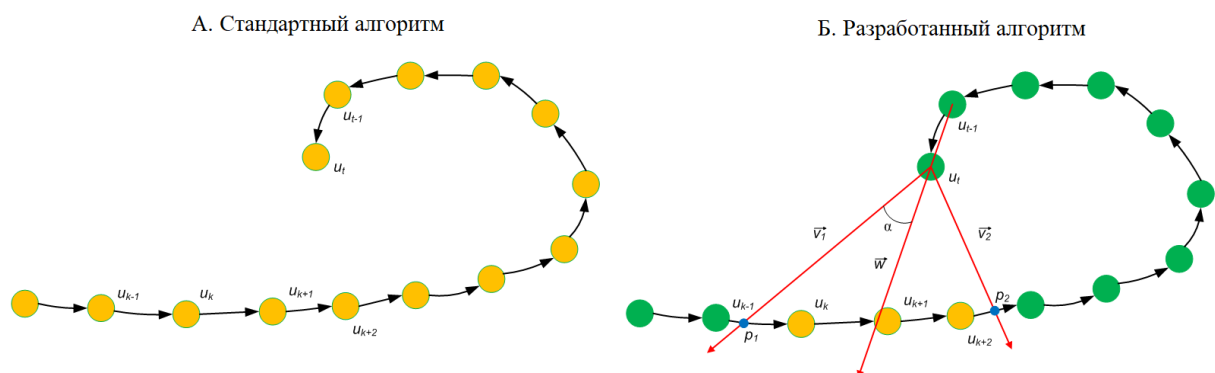


Рис. 2. Ограничение области поиска замыканий траекторий.

Рассмотрим 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, определяемого с одной стороны вектором движения системы, а с другой – настраиваемым параметром α , при изменении которого можно увеличивать или уменьшать область поиска соответствий (рис. 2.Б). Ограничивающее окно строится 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.

Для тестирования производительности и точности разработанного метода был использован набор программных средств Robot Operating System (ROS), который предоставляет инструменты для исследований алгоритмов, использующихся в области робототехники в целом, и для тестирования методов, решающих задачу mvSLAM в частности.

Исследовались методы ORB-SLAM2 и LSD-SLAM и их модифицированные версии, в которых оригинальный алгоритм поиска замыканий траектории был заменен на предложенный в работе (ORB-SLAM2 + LC и LSD-SLAM + LC соответственно). Для исследования использовались открытые коллекции видео-данных Malaga Dataset 2009, Malaga Dataset 2013, LSD-SLAM Dataset, KITTI Dataset. В общей сложности алгоритмы были исследованы на 41 видеопоследовательности. Минимальная длина траектории среди всех видеопоследовательностей – 15.3 м., максимальная – 13 069 м. 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. Основные результаты представлены в таблице 1.

Таблица 1. Результаты экспериментального исследования методов ORB-SLAM2, LSD-SLAM и их модификаций.

| | ORB-SLAM 2 | | LSD-SLAM | |
|---------------------------|------------|------------------|----------|------------------|
| | Оригинал | Модифицированный | Оригинал | Модифицированный |
| Среднее TE (%) | 3.85 | 3.58 | 2.99 | 2.75 |
| Среднее RE (deg/m) | 0.10 | 0.09 | 0.09 | 0.08 |
| Среднее ATE | 0.85 | 0.77 | 0.75 | 0.70 |
| Среднее RPE | 0.88 | 0.81 | 0.85 | 0.81 |
| Среднее CC | 1.32 | 1.23 | 1.40 | 1.32 |
| Среднее время работы (мс) | 0.321 | 0.23 | 0.40 | 0.27 |

Вычислительный эксперимент показал, что среднее время поиска замыкания (обработки одного кадра) уменьшилось на 33% по сравнению с оригинальными методами, а точность построения карты и траектории увеличилась на 7% и 8% соответственно. В результате удалось увеличить качество и быстродействие методов, решающих задачу mvSLAM.

Третья глава описывает разработанную искусственную нейросетевую модель восстановления глубины по изображению с последующей интеграцией ее в методы, решающие задачу mvSLAM, и тестирование на вычислителе Nvidia Jetson TX2 (в том числе – в составе робототехнической системы).

Типовые модели искусственных нейронных сетей для решения задачи восстановления глубины изображения состоят из двух основных модулей: кодировщик и декодировщик. Кодировщик преобразует входящее изображение в набор высокоуровневых особенностей, а декодировщик преобразует полученный набор особенностей в одноканальное изображение (карту глубины). Основные операции кодировщика можно описать следующим образом. Пусть определен тензор размера $H \times W \times C$: $T = \{p_{hwc} \mid p_{hwc} \in \mathbb{R}, h \leq H, w \leq W, c \leq C\}$. Пусть задано ядро свертки $K = \{k_{ijl} \mid k_{ijl} \in \mathbb{R}, i \leq H', j \leq W', l \leq C'\}$. 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, ReLU и т.д.). Одним слоем искусственной нейронной сети будем называть любую из определенных функций. Блоком кодировщика называется комбинацию слоев. Например: $(F_{act} \circ F_{pool} \circ F_{conv_1} \circ \dots \circ F_{conv_N})(T, K) = T', T'$ - набор особенностей, \circ - операция композиции. Архитектурой модели искусственной нейронной сети будем называть комбинацию слоев и блоков.

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 $D = \{d_{ijk}\}$, $i \leq H, j \leq W, k = 1$.

В качестве основного кодировщика в работе было предложено использовать ResNet50, который является одним из наиболее распространенных и эффективных кодировщиков для выделения набора особенностей изображений, использующимся в широком классе задач компьютерного зрения. Несмотря на то, что архитектура состоит из большого количества блоков, ее быстродействие достаточно для работы в реальном времени. 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**. В работе также была исследована версия архитектуры ResNet50 без последних трех сверточных блоков. Выход сети имеет размер $30 \times 40 \times 1024$. Облегченная архитектура работает гораздо быстрее исходной, при этом предполагается, что потери в точности будут незначительные. Облегченная версия архитектуры обозначена как **Lite Basic**.

Первая модификация декодировщика, исследованная в работе, состоит в использовании блоков с разверточными (**Deconv**) слоями 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: $F_{act}(F_{conv}^{-1}(T, K'))$. Подход является базовым для большинства современных декодировщиков, применимых к задаче

восстановления глубины, и он был использован как основной для сравнения предложенных модификаций.

Во второй модификации декодировщика каждый блок содержит слой повышения дискретизации с последующей серией одномерных сверток (non-bottleneck 1D convolution), которая использовалась ранее в сети ERFNet. Повышение дискретизации происходит методом интерполяции ближайшего соседа (F_{int}), затем выполняются два последовательных набора сверток с ядром размера 3×1 и 1×3 и нормализация (batch normalization). Блок архитектуры декодировщика выглядит следующим образом: $(F_{act} \circ F_{norm} \circ F_{int})(F_{conv}(F_{conv}(T, K_1), K_2))$. Выход активируется функцией ReLU. Модификация обозначена как **Upsampling + nonbt**.

Третья модификация – использование блоков сверточного повышения дискретизации (**up-convolution**) в декодировщике. Блок состоит из серии операций повышения дискретизации и сверток: $(F_{act} \circ F_{conv} \circ F_{pool}^{-1})(T, K)$. Предложенный подход работает быстрее, чем **deconv** при сравнимой точности. Для дальнейшего уменьшения времени обработки изображения было предложено заменить слои $F_{conv} \circ F_{pool}^{-1}$ в декодировщике на эквивалентную реализацию с помощью функции **чередования (interl)**. Основная идея заключается в том, что исходное ядро свертки разбивается на четыре производных ядра K_1, K_2, K_3, K_4 размером 3×3 , 2×3 , 3×2 и 2×2 соответственно, происходит свертка исходного тензора с полученными ядрами, а затем - объединение четырех тензоров в один с помощью операции чередования и выполняется активация. Результирующий блок выглядит следующим образом: $(F_{act} \circ F_{interl})(F_{conv}(T, K_1), F_{conv}(T, K_2), F_{conv}(T, K_3), F_{conv}(T, K_4))$

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

Функция ошибки. В процессе тестирования разработанных архитектур моделей искусственных нейронных сетей было установлено, что ошибка на близких расстояниях больше, чем средняя ошибка по всей предсказанной карте глубины. Для задачи навигации робототехнических систем точное определение близких расстояний является критическим, поскольку при неправильной оценке расстояний до близких объектов выше вероятность столкновения. Для уменьшения ошибки на близких расстояниях в процессе обучения обычно используются относительная функция потерь (REL) и функция потерь Huber (и

ее модификация BerHu). Эксперименты с вышеописанными функциями ошибки показали, что они обладают недостатками: функции REL не позволяет значительно повысить точность восстановления глубины на малых расстояниях, поскольку она влияет на весь диапазон восстановленных глубин, а в функциях Huber и BerHu порог k , который отвечает за чувствительность функции к ошибке на определенных расстояниях, настраивается вручную.

В ходе экспериментов с обучением моделей искусственных нейронных сетей был предложен адаптивный (aBerHu) подход для вычисления порогового значения k . На каждом шаге обучения определяется окно $[k - \delta; k]$ и $[k; k + \delta]$, 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 $D_{ij} \in [k; k + \delta]$. Сумма ошибок записывается в R_{upper} . Далее, значения R_{lower} сравнивается с R_{upper} , и обновляется пороговое значение k :

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

Наряду с функцией aBerHu используется комбинация функций среднеквадратичной функции (MSE) и относительной функции (REL) как быстрый аналог (в контексте обучения) оригинальным функциям, позволяющий учесть ошибку на близких расстояниях.

В процессе обучения был выявлен значительный недостаток – архитектуры моделей искусственных нейронных сетей с блоком **чередования** в декодирующей обучаются гораздо медленней, чем архитектуры с оригинальным блоком **up-convolution**. Поскольку оба блока (сверточного повышения дискретизации и свертки с последующим чередованием) являются эквивалентными, то было предложено использовать блок **сверточного повышения дискретизации** для обучения сети, а затем обученные веса переносить в модификацию сети с операцией чередования (**T**). Таким образом удастся получить прирост производительности без потери качества восстановления глубины.

Разработанные архитектуры были исследованы на встраиваемой платформе Nvidia Jetson TX2, которая обладает компактными размерами и пониженным энергопотреблением, что позволяет использовать ее в качестве бортового вычислителя робототехнической системы. Архитектуры были реализованы с помощью библиотек Tensorflow и Keras на языке программирования Python. Операции чередования и сверточного повышения дискретизации были реализованы для центрального (CPU) и графического (GPU) процессоров на языках программирования C/C++ с помощью компиляторов g++

и nvcc соответственно. Для обучения был использован гибридный высокопроизводительный вычислительный комплекс ФИЦ ИУ РАН.

На этапе обучения использована открытая коллекция данных NYU Dataset v2, которая состоит из более чем 400 000 пар изображение-глубина, заснятых на более 470 разных сценах.

Результаты тестирования описанных в работе архитектур представлены в таблице 2.

Таблица 2. Результаты тестирования архитектур восстановления глубины изображений

| Архитектуры | Функция потерь | Разрешение | Декодировщик | MSE | REL | Время на ПК (с) | Время на Jetson (с) |
|----------------------------|-------------------|------------|-------------------|--------------|--------------|--------------------|------------------------|
| Wang et al. | - | - | - | 0.555 | 0.220 | - | - |
| Eigen et al. | - | 304x228 | - | 0.823 | 0.215 | - | - |
| Laina et al. | BerHu | 304x228 | - | 0.328 | 0.127 | - | - |
| Alhashim et al. | - | 640x480 | - | 0.152 | 0.103 | - | - |
| Basic | BerHu | 640x480 | Deconv | 0.467 | 0.186 | 0.144 | 0.152 |
| Basic + SC | BerHu | 640x480 | Deconv | 0.487 | 0.194 | 0.521 | 0.563 |
| Basic + SC | aBerHu | 640x480 | Upsampling.+nonbt | 0.440 | 0.184 | 0.158 | 0.215 |
| Basic + SC | MSE+REL | 640x480 | Upsampling.+nonbt | 0.419 | 0.173 | 0.158 | 0.215 |
| Basic + SC | MSE+REL | 320x240 | Upsampling.+nonbt | 0.408 | 0.180 | 0.049 | 0.062 |
| Lite Basic + SC | MSE+REL | 320x240 | Upsampling.+nonbt | 0.533 | 0.202 | 0.035 | 0.049 |
| Basic + SC + interl | MSE+REL | 640x480 | Up-convolution. | 0.514 | 0.206 | 0.285 | 0.328 |
| Basic + SC + interl | MSE+REL | 320x240 | Up-convolution | 0.580 | 0.215 | 0.057 | 0.067 |
| Basic + SC + interl + T | MSE+REL | 640x480 | Up-convolution | 0.445 | 0.178 | 0.181 | 0.227 |
| Basic + SC + interl + T | MSE+REL | 320x240 | Up-convolution | 0.495 | 0.181 | 0.048 | 0.061 |
| Lite Basic + interl + T | MSE+REL | 640x480 | Up-convolution | 0.658 | 0.233 | 0.101 | 0.135 |
| Lite Basic + interl + T | MSE+REL | 320x240 | Up-convolution | 0.660 | 0.236 | 0.027 | 0.037 |

В первой колонке таблицы – названия архитектур, во второй – используемые функции потерь для обучения. Разрешение показывает высоту и ширину входящего изображения для конкретной архитектуры. Далее – название блока декодировщика, значение ошибок MSE и REL (меньше – лучше). Последние две колонки показывают время обработки одного изображения на ПК и на Nvidia Jetson TX2. Лучший результат в плане быстродействия на Nvidia Jetson TX2 показала архитектура Lite Basic + interl + T (37 мс).

С другой стороны, в архитектурах Basic + SC + Upsampling + nonbt удалось добиться лучшей относительной ошибки REL, чем в некоторых архитектурах из

смежных работ зарубежных авторов, которые не нацелены на работу в реальном времени. Один кадр обрабатывается 62мс, что достаточно для использования в задаче mvSLAM. Наиболее подходящей по времени работы и качеству является архитектура Basic + SC + Upsampling + nonbt. Она в дальнейшем использовалась для работы на реальной робототехнической системе.

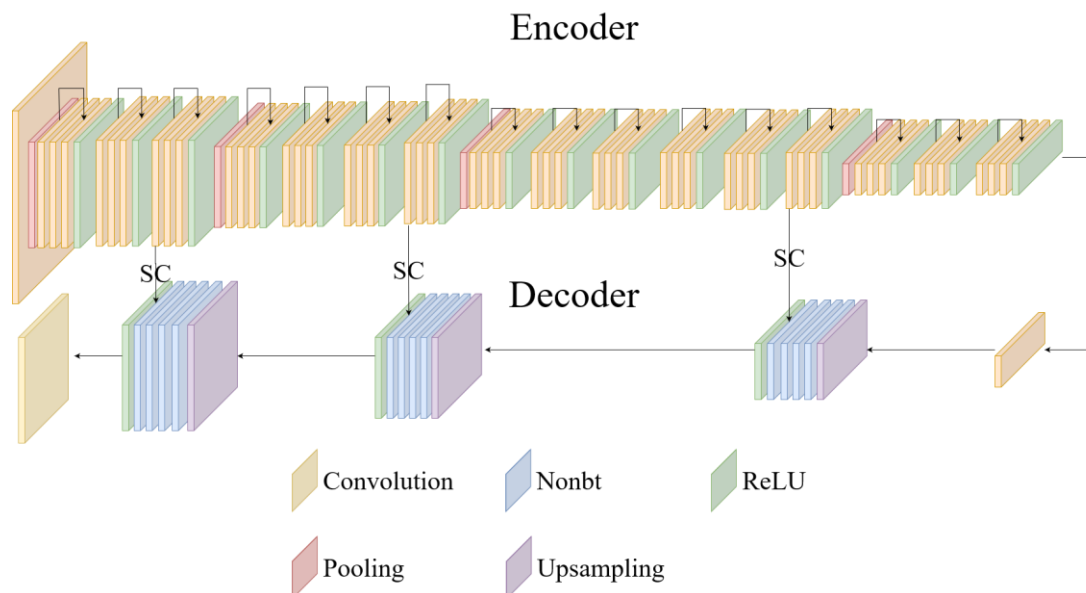


Рис. 3. – Реализованная на встраиваемой платформе архитектура.

Для исследования применимости разработанной модели к задаче mvSLAM реализован модуль (узел) для набора программных инструментов Robot Operating System (ROS). Для картирования и локализации использован открытый узел RTAB-Map. Все вычисления выполняются автономно на Nvidia Jetson TX2 и на реальной робототехнической системе. Робототехническая система – колесный робот, оснащенный Nvidia Jetson TX2 в качестве бортового вычислителя, Raspberry Pi в качестве платы управления двигателями и камерой. Управление роботом производилось дистанционно с рабочей станции. Эксперимент проводился в здании робототехнического центра ФИЦ ИУ РАН. Задачей являлся проезд по замкнутому коридорному помещению без потери локализации. Эксперименты показали, что разработанная модель позволяет точно и быстро производить восстановление карты и вычисление позиций системы на встраиваемой платформе. Архитектура нейронной сети работает на графическом процессоре со скоростью 16 кадров в секунду, в то время как алгоритм mvSLAM – на центральном, поэтому алгоритмы не занимают ресурсы друг у друга.

В четвертой главе описана разработанная модель для оценки качества картирования с учетом особенностей задачи mvSLAM.

Для количественного сравнения методов, решающих задачу mvSLAM, необходимо использовать специальные метрики как для измерения точности построенной траектории, так и для оценки качества результирующей карты. Измерение качества локализации происходит независимо от картирования. В случае задачи mvSLAM, результатом работы (выходными данными) локализации является траектория, которая состоит из последовательности позиций робота (камеры) в среде. Каждой позиции соответствуют координаты в трехмерном пространстве и вектор направления.

Оценим качество работы модуля локализации. Пусть дана последовательность трехмерных координат или позиций:

$$U = \{u_t = (p_t, q_t), t \in [1; T], T \in N\}$$

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

2) $RPE(U, f) = \sum_{t=2}^m \|q_t^{-1} \otimes (p_t - p_{t-1}) - f(0, (q_t)^{-1}) \otimes (f(p_t, 0) - f(p_{t-1}, 0))\|^2$ – 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 методов, решающих задачу SLAM (и mvSLAM, в частности). Метрика АТЕ позволяет оценить качество построения позиций на всей протяженности траектории, в то время как RPE оценивает смещения между соседними позициями. Задача 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.

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 сравнения между собой двух множеств трехмерных точек. На рис. 4 показаны: А) 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).

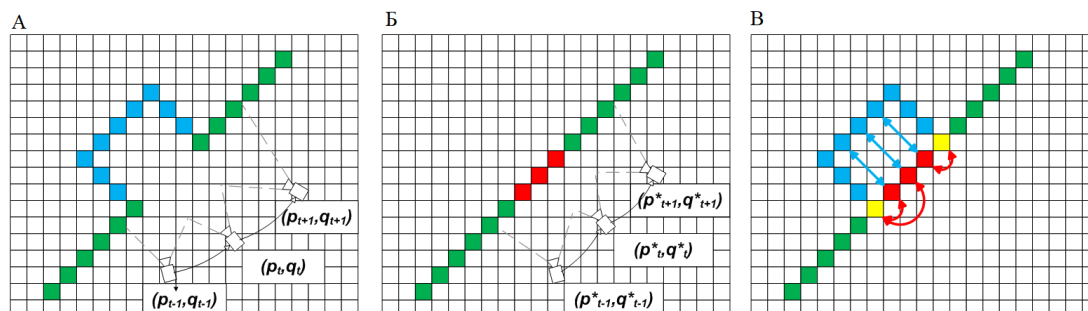


Рис.4. Визуализация проблемы сравнения карт стандартными функциями соответствия.

Базовым подходом для нахождения соответствия точек двух множеств является поиск ближайших соседей после приведения двух множеств к единой системе координат. Однако, в задаче mvSLAM, одной из особенностей сопоставления методом ближайшего соседа является частая ошибка при картировании дверных проемов в коридорной местности, когда сам проем наносится как препятствие на карту (рис. 4. Б), в то время как на этом месте должно быть пустое пространство с препятствием за ним (рис. 4. А). При использовании сопоставления методом ближайших соседей, красные точки, нанесенные на карту методом, решающим задачу mvSLAM 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).

Описанная выше задача требует разработки особой модели сопоставления точек двух множеств с учетом контекста задачи одновременного картирования и локализации по видеопотоку. В качестве связующего элемента между двумя множествами трехмерных точек, предлагается использовать позиции, которые были вычислены в ходе реконструкции трехмерной карты из двумерного изображения.

Задача поиска соответствий при сопоставлении двух облаков точек может быть записана следующим образом. Пусть даны истинная карта M и карта, построенная в ходе mvSLAM карта M^* :

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

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

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^* - истинные позиция и ориентация и рассчитанные позиция и ориентация камеры.

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 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:

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

Функция соответствия выглядит следующим образом: $f(m_i^*) = p_t + \alpha r_t$. Т.е. вектор, указывающий из камеры на точку в истинной карте переносится на предсказанную карту и масштабируется до тех пор, пока не попадет в предсказанную точку, которая и будет считаться соответствием. 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. 5.

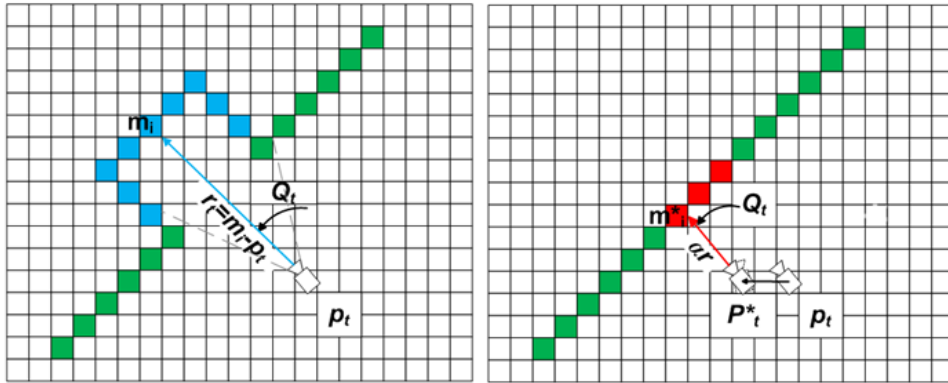


Рис. 5. Пример расчета соответствия.

С помощью предлагаемой функции предлагается ввести следующие метрики:

$AME(M^*, f_{ME}) = \sqrt{\frac{1}{N} \sum_{t=1}^N \|f_{ME}(m_i^*) - m_i^*\|^2}$ - Absolute Mapping Error (AME), где $N = \min(|M|, |M^*|)$. Метрика рассчитывает среднее расстояние

между точками, для которых установлено соответствие с помощью введенной функции f_{ME} .

$$RME(M^*) = \sqrt{\frac{1}{n} \sum_{t=1}^N \left\| M_{q_t}(m_i - p_t) - M_{q_t^*}(m_i^* - p_t^*) \right\|^2} - \text{Relative Mapping}$$

Error (RME). RME учитывает средние относительно текущей позиции расстояния до точек на карте с учетом установленного соответствия.

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. Финальные значения предложенных метрик AME и RME зависят от выбора момента времени t , при котором будет установлено соответствие между точками на истинной и построенной картах.

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 расстоянию до точки, 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.

Для проверки предложенной модели оценки качества картирования был исследован алгоритм RTAB-Map. В качестве экспериментальной среды был выбран фотореалистичный симулятор Habitat. На вход алгоритму подавались изображения и истинные карты глубин (RTAB-Map + Habitat depth), карты глубин, предсказанные моделью искусственной нейронной сети, описанной в главе 2 (RTAB-Map + FCNN). Результаты расчета метрик отклонения траекторий ATE, RPE, среднеквадратичное отклонение с методом ближайших соседей

(AME(NN)), среднеквадратичное (AME(ours)) и относительное (RME(ours)) отклонения с разработанным методом сопоставления, представлены в таблице 3.

Таблица 3. Результаты оценки качества картирования и локализации.

| Название | ATE | RPE | AME(NN) | AME(ours) | RME(ours) |
|--------------------------|-------|-------|---------|-----------|-----------|
| 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 AME(NN) у алгоритма RTAB-Map + FCNN в 9 раз больше, чем у RTAB-Map с известными картами глубин, а по метрикам AME(ours) и RME(ours) – примерно в 2.7 раза больше. То есть разница в качестве построенных карт в контексте задачи mvSLAM 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. Полученные данные свидетельствуют о том, что разработанная функция сопоставления позволяет точнее оценивать качество картирования по сравнению со стандартными функциями.

В заключении приведены основные результаты работы:

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.

Публикации автора по теме диссертации

В изданиях из списка ВАК РФ

1. 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.
2. Боковой А.В. Архитектура системы автоматического управления группой малых беспилотных летательных аппаратов // Информационные технологии и вычислительные системы. – 2018. – №. 1. – С. 68-77.

В изданиях, индексируемых Scopus/WoS:

1. Muravyev K., Bokovoy A., Yakovlev K. tx2_fcnn_node: An open-source ROS compatible tool for monocular depth reconstruction // SoftwareX. – 2022. – P. 1-6.
2. 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.
3. Bokovoy A., Muraviev K. Assessment of Map Construction in vSLAM // 2021 International Siberian Conference on Control and Communications (SIBCON). – IEEE, 2021. – P. 1-6.
4. 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.
5. 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.

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., 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.
8. 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.
9. 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.

В трудах конференций:

1. Муравьев К.Ф., Боковой А.В., Яковлев К.С. Оценка качества алгоритмов картоирования 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.
2. 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.
3. 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.

4. 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.
5. 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.
6. 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.