Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа Прикладной Математики и Информатики
Научно-образовательный центр когнитивного моделирования

Направление подготовки / специальность: 03.04.01 Прикладные математика и физика
Направленность (профиль) подготовки: Методы и технологии искусственного интеллекта

ПОСТРОЕНИЕ ТРЕХМЕРНЫХ СЕМАНТИЧЕСКИХ КАРТ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

(магистерская диссертация)

Студент:
Безуглый Виталий Дмитриевич

_____
(подпись студента)

Научный руководитель:
Юдин Дмитрий Александрович,
канд. техн. наук

_____
(подпись научного руководителя)

Консультант *(при наличии)***:**
_____
_____
(подпись консультанта)

Москва 2024

Содержание

1 Введение **3**

2 Задача картирования в робототехнике **6**

2.1 Типы трехмерных карт 6

2.2 Математические модели, используемые при реконструкции трехмерных карт 7

2.3 Карты, содержащие информацию о семантике сцены 13

2.4 Метрики и оценка качества карты 14

2.5 Обзор современного состояния проблемы 16

2.6 Постановка задачи и цель работы 23

3 Семантически обусловленные алгоритмы реконструкции трёхмерной меш-карты по размеченным лидарным данным **25**

3.1 Архитектура системы реконструкии трёхмерных семантических карт с использованием явных представлений (TSDF) 25

3.2 Архитектура системы реконструкии трёхмерных семантических карт с использованием неявных обучаемых представлений 26

3.3 Функции потерь 31

3.4 Оценка качества семантической меш-карты 32

3.5 Детали реализации 33

4 Результаты тестирования разработанного метода картирования **36**

4.1 Описание используемых наборов данных 36

4.2 Подготовка данных 40

4.3 Результаты экспериментов 42

4.4 Сравнение результатов с существующими методами 44

4.5 Выводы 52

5 Заключение **54**

Аннотация

Алгоритмы реконструкции трехмерных семантических карт, являются важным элементом бортовых систем компьютерного зрения транспортных средств. Такие карты позволяют строить среды симуляции и решать задачи, для которых необходимо полное понимание сцены. В данной работе предложен подход семантически обусловленной реконструкции трёхмерных семантических карт и реализован алгоритм картирования Sea-SHINE, учитывающий семантику сцены. Предложенный подход позволяет восстанавливать трёхмерные семантические карты учитывая семантику сцены, а не просто добавлять её как атрибут. В частности это заметно по качеству реконструкции небольших элементов сцены, таких как дорожные знаки и столбы. Разработанный алгоритм протестирован на публичных наборах данных и оценены при помощи разработанного метода оценки качества. Полученные результаты показывают, что предложенный алгоритм может восстанавливать трехмерные семантические карты на уровне современных работ, при этом качественнее реконструируя небольшие элементы сцены.

1 Введение

Задача реконструкции трёхмерной карты местности является одной из ключевых в области мобильной робототехники и беспилотного транспорта. Такие карты необходимы мобильным роботам для локализации себя в пространстве, а также могут использоваться в различных приложениях, таких как распознавание объектов, понимание сцены и моделирование. Семантические карты - карты, включающие в себя семантическую информацию - о типах объектов представленных на сцене, в свою очередь являются естественным продолжением идеи картирования и позволяют качественнее решать задачи восприятия окружающей среды и взаимодействия с ней.

Реконструкция трёхмерных карт возможна в различных форматах, таких как облака точек, меши, воксельные сетки и другие. Представление карты в виде меша (трёхмерной полигональной сетки) является одним из популярных и удобных способов представления трёхмерной информации. Меш карты являются наиболее полными моделями среды, так как они содержат информацию о геометрии и топологии сцены, а также позволяют хранить дополнительные атрибуты, такие как цвет, нормали, текстуры и семантическую информацию. В то же время меш является сложным и требовательным к вычислительным ресурсам представлением, что делает задачу их реконструкции актуальной и сложной.

Современные подходы к реконструкции трёхмерных карт местности в виде меша можно разделить на 2 основные категории: аналитические и нейросетевые. Аналитические методы представляют собой круг подходов, связывающих математические модели представления трёхмерных поверхностей и численных методов, позволяющих укладывать поток реальных данных в такие модели. Они позволяют получать высококачественные результаты и часто могут работать в реальном времени, но чувствительны к шумным данным и ограниченны рамками своей модели. Нейросетевые методы, в свою очередь, являются более гибкими и также позволяют строить качественные карты, но требуют больше вычислительных мощностей и редко могут работать в реальном времени. В последнее время среди нейросетевых подходов наиболее популярны алгоритмы, восстанавливающие карту в виде неявных обучаемых представлений, из-за чего методы реконструкции трехмерных карт часто разделяют на явные (аналитические) и неявные (нейросетевые) методы реконструкции.

Методы, использующие обучаемые неявные представления представляют наи-

больший интерес. Принцип работы заключается в представлении карты как набора участков поверхности, каждый из которых описывается некоторым обучаемым представлением - численной переменной. Реконструкция карты производится при помощи небольшой нейросети - декодера. Неявные представления - переменные - подаются на вход декодеру, который восстанавливает геометрическую информацию, «записанную» в этом представлении. При реализации таких методов, переменные описывающие карту создаются при помощи фреймворков, поддерживающих автоматическое дифференцирование (например, PyTorch или TensorFlow), что позволяет применять к ним градиентные методы оптимизации. Таким образом карта представляет собой своего рода большую распределенную нейросеть, которая «помнит» геометрию среды и специальная нейросеть - декодер может эту информацию восстановить. Реконструкция карты представляет собой обучение этих двух нейросетей на данных, полученных с датчиков робота. Представления названы неявными, так как они не являются результатом применения некоторой функции к входным данным, а возникают в процессе численного решения своего рода «обратной задачи» в контексте реконструкции поверхности.

Эти подходы позволяют получать компактное и информативное представление карты, которое за счёт своей природы может быть использовано для любых задач, требующих трёхмерной информации о местности и использующих нейросетевые методы. Однако, такие методы на данный момент уступают аналитическим в скорости работы и местами точности реконструкции.

В обеих группах методов есть примеры, реконструирующие «семантические карты» - карты, которые помимо геометрической информации содержат и семантическая информация о местности. Каждый элемент карты (вершина полигональной сетки или сам полигон) помечен меткой, соответствующей классу объекта, которому он принадлежит. И общим недостатком всех существующих методов является подход к учету семантики - семантическая информация представляется просто как дополнительный атрибут, и не участвует в процессе реконструкции. Редкие алгоритмы одновременного картирования и локализации (SLAM) используют семантические метки для улучшения решения задачи визуальной одометрии или замыкания циклов, но ни одна работа не предлагает метода, который бы учитывал семантическую информацию в процессе реконструкции.

В Разделе 2 будут введены основные понятия, необходимые для понимания постановки задачи, приведен обзор существующих методов реконструкции трёхмерных

семантических карт, введены метрики используемы для оценки качества таких карт и сформулирована формальная постановка задачи.

В Разделе 3 будет рассмотрен предложенные подходы реконструкции трёхмерных семантических карт, описана их архитектура, новизна и предложенные улучшения.

В Разделе 4 будут приведены используемые наборы данных, описаны эксперименты, проведенные для оценки качества предложенных методов, приведены результаты их работы, сравнение с существующими методами и анализ полученных результатов.

В результате выполнения научно-исследовательской работы были получены следующие результаты:

- Предложен подход семантически обусловленной реконструкции трёхмерных семантических карт на основе неявных обучаемых представлений, а также реализован алгоритм реализующий такой подход
- Предложена функция потерь и метод формирования обучающей выборки, учитывающие семантику сцены
- Разработан подход и утилита оценки качества трёхмерных семантических карт
- Собран и размечен мультимодальный, мультисезонный набор данных
- Проведены эксперименты с предложенными подходами и архитектурами, и проведен их анализ

Полученные результаты показывают, что предложенный подход решает поставленную задачу реконструкции небольших элементов сцены. Разработанный алгоритм позволяет восстанавливать трёхмерные семантические карты по уровню реконструкции соответствующие современным работам, хотя и имеет некоторые ограничения и перспективу для улучшения.

2 Задача картирования в робототехнике

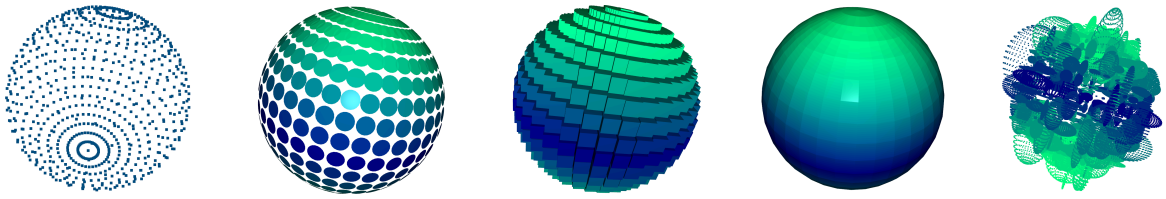
В данном разделе работы будут введены необходимые понятия, определены метрики качества, произведен обзор существующих решений проблемы, а также дана формальная постановка задачи реконструкции трехмерной семантической карты.

В разделе 2.1 будут рассмотрены основные типы трехмерных карт, используемых в робототехнике. В разделе 2.2 - математические модели и алгоритмы, используемые при реконструкции трехмерных карт. В разделе 2.4 - метрики качества трехмерных карт. В разделе 2.3 дано определение семантическим картам. В разделе 2.5 - обзор и анализ существующих решений проблемы. В разделе 2.6 - формальная постановка задачи и цель работы.

2.1 Типы трехмерных карт

Существует несколько основных способов представления 3D карты:

1. Трехмерное облако точек (Рис. 1 (а)).
2. Сурфели (surfels) - направленные участки поверхности (Рис. 1 (б)).
3. Трехмерная решетка (Рис. 1 (в)) - карта представляет собой упорядоченный набор ячеек, центры которых являются узлами 3D прямоугольной сетки с фиксированным шагом.
4. Mesh (трехмерная полигональная сетка) (Рис. 1 (г)).
5. Гауссово рассеивание (Gaussian Splatting) (Рис. 1 (д)).



(а) Трехмерное облако точек (б) Направленные участки поверхности (в) Трехмерная клеточная решетка (г) Полигональная сетка - меш (д) Гауссово рассеивание

Рис. 1: Иллюстрация различных моделей представления данных

2.2 Математические модели, используемые при реконструкции трехмерных карт

Euclidean Signed Distance Field (ESDF) - математическая модель, которую удобно использовать в качестве промежуточного представления данных при реконструкции трехмерных карт.

Пусть S - некоторая ориентированная поверхность в трехмерном пространстве $V \subset \mathbb{R}^3$, функция F имеет вид:

$$F(p) = \min_{p' \in S} \|p - p'\| \cdot (\vec{n'_S}, \vec{p}) \tag{2.1}$$

где $p \in V$, $\vec{n'_S}$ - нормаль к поверхности S в точке p', а операция $\|p - p'\|$ - задает Евклидово расстояние между точками p, p' (см. Рис. 2).

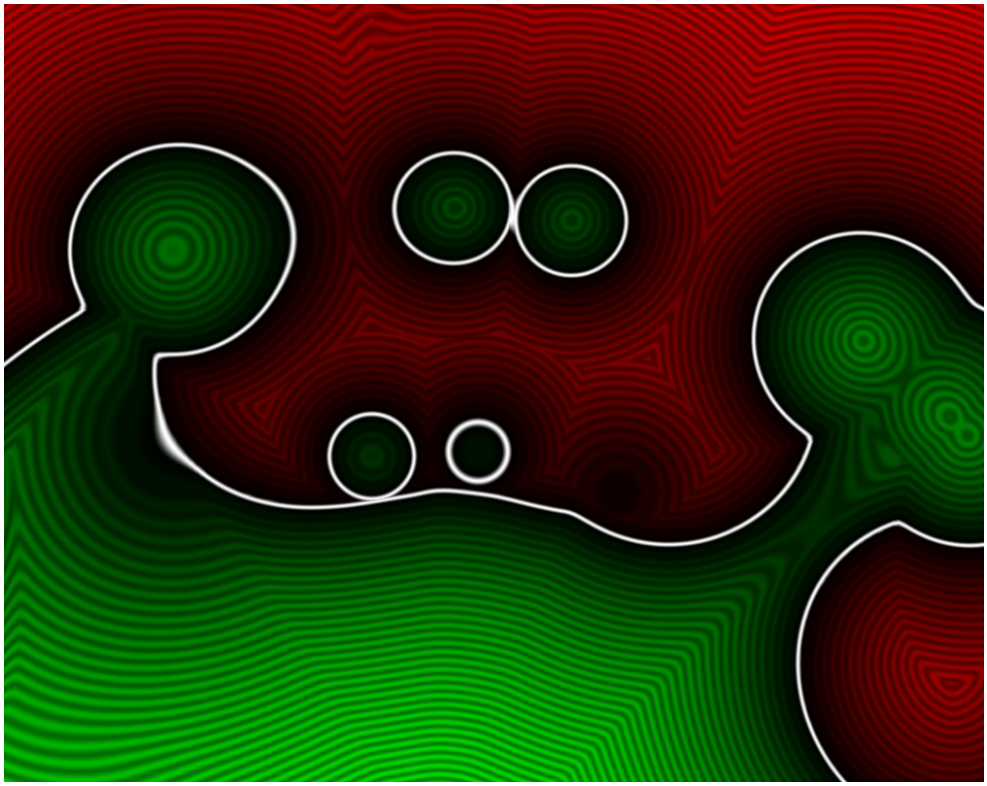


Рис. 2: Euclidean Signed Distance Field (2D) - красный и зеленый цвета символизируют различные арифметические знаки, интенсивность цвета соответствует модулю расстояния, белым цветом обозначена сама поверхность.

Тогда говорят что функция F задает поле Евклидовых расстояний со знаком (ESDF) на пространстве V.

На практике часто используется дискретная версия функция F, которая задается трехмерной решеткой (см. раздел 2.1) в ячейках (или узлах решетки, в за-

висимости от формулировки) хранится усредненное значение F для всего объема занимаемого ячейкой (см. Рис. 3).

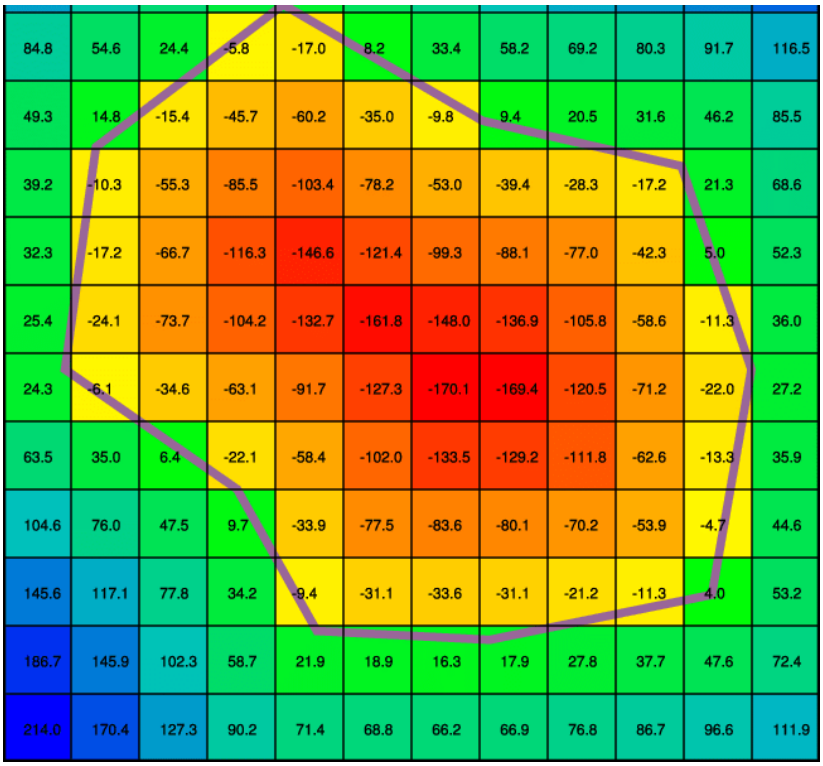


Рис. 3: Euclidean Signed Distance Field (2D) - в дискретном виде, в каждой ячейке указано усредненное значение расстояния до поверхности. Ячейки раскрашены в соответствии с диапазоном расстояний.

При построении карты данные с сенсоров используются для формирования такого поля, на заданном объеме, после чего при помощи алгоритма марширующих кубов (см. п. 2.2) восстанавливается непосредственно поверхность. Удобство такого представления карты в том, что оно не требует оптимизации и может обновляться локально по мере поступления новых данных.

Данная модель не используется в прикладных работах, ввиду существования более оптимальной версии описанной ниже.

Truncated Signed Distance Field (TSDF) - то же поле Евклидовых расстояний со знаком, но упрощенное для ускорения реконструкции.

Реконструкция ESDF предполагает задание функции расстояния в каждой ячейке выделенного объема. Функция расстояния нам нужна для восстановления поверх-

[1]Источник [1]

ности, для которой достаточно знать значения функции в непосредственной близости от самой поверхности (см. п. 2.2), тогда как большую часть трехмерной карты занимает пустое пространство.

Ввиду этого на практике при интеграции новых данных с сенсоров (трехмерных облаков точек) - функция расстояния записывается только в ячейки сетки находящиеся не далее чем некоторое пороговое значение от поверхности, это пороговое значение является гиперпараметром такого подхода (см. Рис. 4).

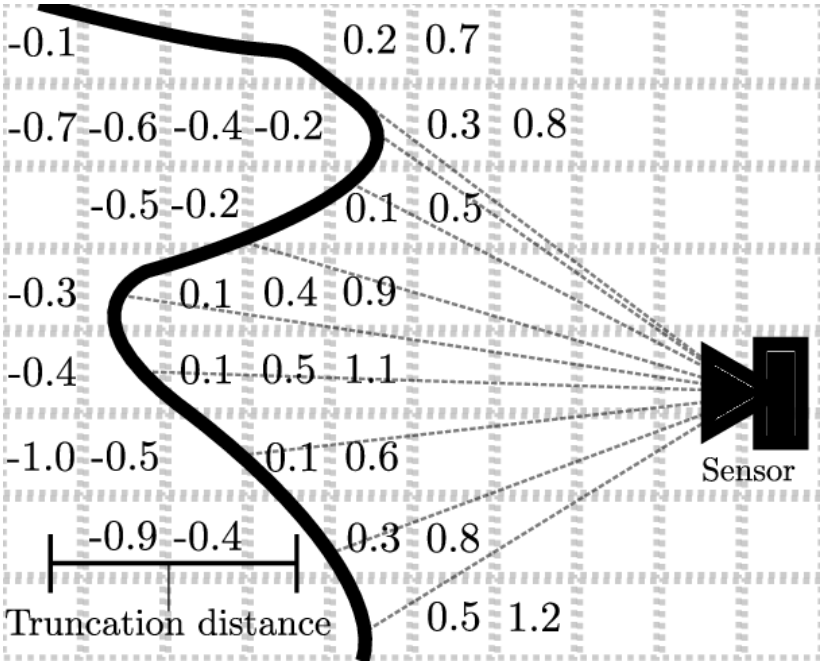


Рис. 4: Truncated Signed Distance Field (2D) - в дискретном виде, в каждой ячейке указано усредненное значение расстояния до поверхности. При этом расстояние хранится только для ячеек ближе некоторого порогового значения (truncation_distance) от поверхности.

Поверхность восстановленная по такому «укороченному» полю расстояний получается менее точной, так как часть информации все же теряется, но такой подход позволяет сильно экономить время интеграции новых данных. Настолько, что его можно использовать в алгоритмах, работающих в реальном времени.

Данная модель используется в таких работах как [32, 35, 10].

Screened Poisson Surface Reconstruction (SPSR) [4] - подход к реконструкции поверхности по набору ориентированных точек в трехмерном пространстве через решение уравнения Пуассона.

Как и в подходе ESDF/TSDF мы пытаемся найти индикаторную функцию, ко-

торая бы задавала нашу поверхность, но если в TSDF мы искали функцию неявно - через поле значений функции расстояния, опираясь на точки как на нули такой функции, то в данном подходе мы наоборот ищем функцию, аппроксимирующую поверхность, в явном виде.

Пусть $\vec{V} \subset \mathbb{R}^3$ - множество ориентированных точек, ∂M - искомая ориентированная поверхность, а χ_M - индикаторная функция нашей поверхности, тогда множество $\vec{V}$ - рассматривать как множество значений градиента индикаторной функции χ_M (см. Рис. 5).

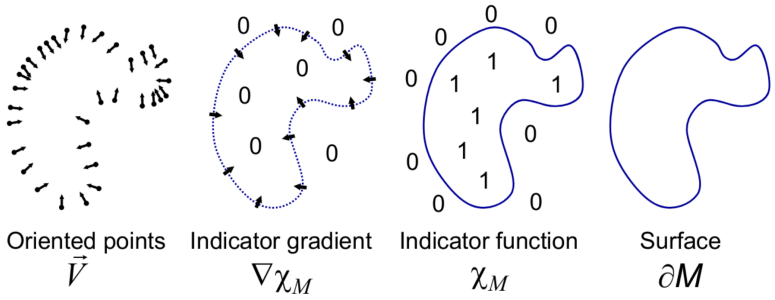


Рис. 5: Иллюстрация постановки задачи реконструкции поверхности через уравнение Пуассона из работы [56]

Таким образом задача реконструкции поверхности может быть сформулирована как аппроксимация функции, по известному множеству значений ее градиента:

$$\min_{\chi_M} \left\| \nabla\chi_M - \vec{V} \right\| \tag{2.2}$$

Которое через применение оператора дивергенции ∇ к обеим частям уравнения сводится к решению уравнения Пуассона:

$$\Delta\chi \equiv \nabla \cdot \nabla\chi = \nabla\vec{V} \tag{2.3}$$

Дальнейшее решение сводится к следующим шагам:

1. Построение на исходном множестве точек $\vec{V}$ октодерева (см.Рис. 6)

2. Выбор базиса $\{B_1, \dots B_N\} : \mathbb{R}^3 \to \mathbb{R}$ из B-сплайнов

3. Итеративная по-уровневая (по уровням октодерева) аппроксимация индикаторной функции отдельно в каждой ячейке октодерева.

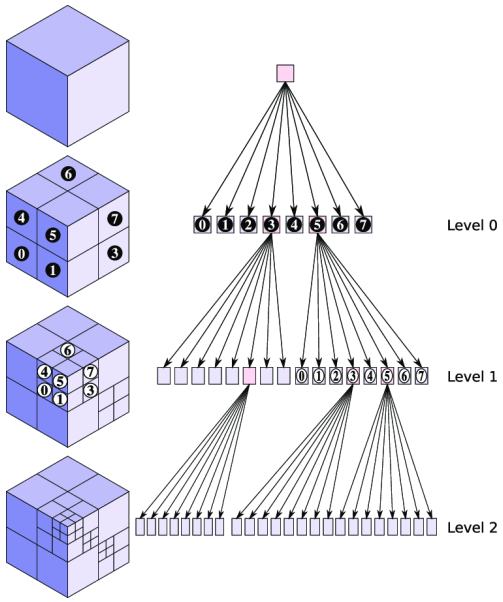


Рис. 6: Иллюстрация структуры данных octotree (октодерево)

Таким образом индикаторная функция для точки p выглядит следующим образом:

$$\chi(p) = \sum_{d=0}^{D} \sum_{i} x_i^d B_i^d(p) \tag{2.4}$$

где D - глубина октодерева, x_i^d - коэффициент i-го базисного сплайна для d уровня дерева.

Далее по полученной индикаторной функции меш реконструируется при помощи алгоритма марширующих кубов 2.2 по тем же ячейкам октодерева.

Такая формулировка задачи и алгоритм её решения были изначально представлены в [56] как подход «Poisson Surface Reconstruction (PSR)». Через некоторое время та же исследовательская группа предложила улучшенную версию подхода к решению [4] под названием «Screened Poisson Surface Reconstruction», который в последствии стал классическим в задаче реконструкции поверхности.

На русский язык «screened» можно перевести как «растеризованный» и ключевое отличие (хоть и не единственное) SPSR от PSR заключается в том, что в PSR оптимизируемая функция задавалась входными точками $\vec{V}$ неявно, и в результате оптимизации функция ошибки стремилась к 0 «в среднем» по всем точкам, то в SPSR был использован подход, который интерполировал исходные точки в явном виде. Это позволило лучше учитывать шум в данных и потенциально взвешивать входные данные исходя из степени «уверенности» в них.

Преимуществом такого подхода является адаптивное и регулируемое (путем вы-

бора параметров октодерева) разрешение, работа с шумными данными и скорость работы. Однако в силу специфики подхода, реконструируемая поверхность стремится быть непрерывной и замкнутой, что приводит к реконструкции поверхности даже там где точек не было, что не всегда соответствует задаче.

Также существенным недостатком подхода является специализация на работе с «полным» набором данных, что усложняет его использование в задаче реконструкции карт местности (но не делает невозможным [19]).

Marching Cubes [57] - подход к реконструкции меша по известной дискретной индикаторной функции.

Постановка задачи следующая: имеем трехмерную решетку в узлах которой заданы значения индикаторной функции поверхности. Хотим восстановить эту поверхность в виде меша.

Для этого рассмотрим ячейки, на вершинах которых индикаторная функция принимает различные по знаку значения. То есть ячейки через которые проходит поверхность. Идея очень простая: у ячейки всего 8 вершин, каждая из которых может принимать 2 значения $\{0, 1\}$ итого имеем всего $2^8 = 256$ комбинаций значений функции на вершинах ячейки. Если учесть симметрию куба (ячейки), то всего окажется 15 уникальных (с точностью до поворота) комбинации. 1 вариант - 1 способ провести поверхность через ячейку (см. Рис. 7).

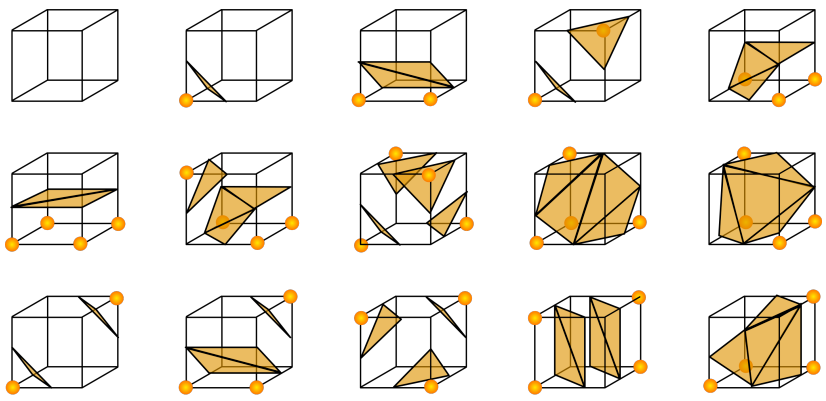


Рис. 7: Иллюстрация 15-ти возможных конфигураций поверхности, используемых в подходе марширующих кубов. Желтыми точками обозначены вершины с отрицательным значением SDF, те что не отмечены - имеют значение больше нуля.

Тогда можно просто рассмотреть все такие варианты, заранее прописать для каждого «оптимальный» вариант проведения поверхности и дальше просто итера-

тивно идти по ячейкам и выбирать нужную конфигурацию исходя из значений индикаторной функции в вершинах ячейки.

В случае использования не просто индикаторной функции, а ESDF/TSDF (когда в вершинах не $\{0, 1\}$, а расстояния до поверхности) теоретически можно рассчитывать такие конфигурации для каждой ячейки исходя из расстояний, но в таком случае подход требует несравнимо больше вычислений, а также возникает проблема согласованности краев поверхности в соседних ячейках, решение которого приводит к еще большей вычислительной сложности. Поэтому для ESDF/TSDF не делается исключений и алгоритм применяется в описанном виде.

2.3 Карты, содержащие информацию о семантике сцены

Семантическая информация сочетает в себе компактность представления и высокую смысловую нагрузку, что делает ее идеальным инструментом для использования в различных задачах робототехники, включая картирование и локализацию. В задаче картирования семантическая информация обычно используется для исключения данных, связанных с динамическими объектами [14, 20, 26], для построения объектно-ориентированной карты [32, 12] или для реконструкции семантически раскрашенных карт [52, 42]. Таким образом, если на карту «наносится» семантическая информация, то карта становится "семантической"(см. Рис. 8).

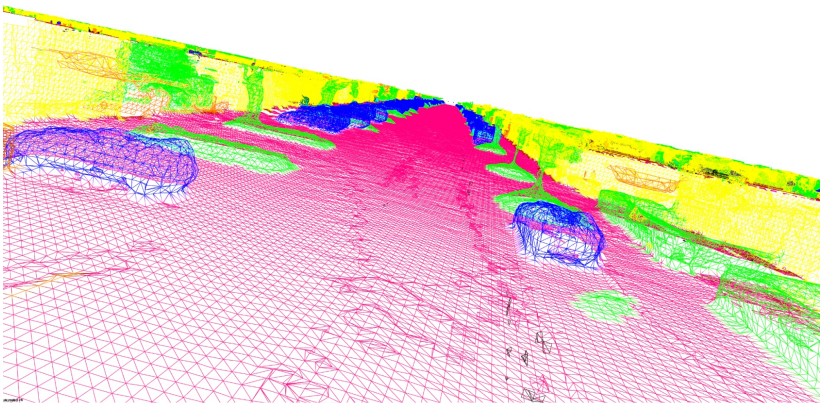


Рис. 8: Иллюстрация трехмерной семантической карты в виде меша. Карта раскрашена в соответствии с семантическими классами.

В то же время способ представления семантической информации на самой карте

достаточно тривиален - в дополнительные атрибуты элемента карты (в зависимости от модели представления) добавляется информация о классе семантической модели или, иногда, конкретном экземпляре семантического класса.

2.4 Метрики и оценка качества карты

Есть 2 основных подхода к оценке качества реконструкции поверхности:

1. Метрики - если есть ground-truth поверхность, с которой можно сравнить результат реконструкции.

2. Качественное (визуальное) сравнение с исходными данными и другими методами - соответственно, если истинной поверхности нет. Например, при реконструкции карты местности.

Сравнение трехмерных поверхностей произвольной формы - очень сложная задача с вычислительной точки зрения, поэтому ее обычно сводят к более простой - задаче сравнения облаков точек. На каждой из сравниваемых поверхностей случайным образом выбирается заранее обусловленное большое число точек (зависит от масштаба поверхности) и считаются метрики соответствия друг другу для полученных облаков точек. Ниже приведены формулы расчета наиболее популярных метрик.

Предварительные расчеты Пусть $G = \{p \in \mathbb{R}^3\}$ - множество точек выбранных на истинной поверхности, $S = \{p \in \mathbb{R}^3\}$ - множество точек выбранных на восстановленной поверхности, и функция d - определяет минимальное расстояние от некоторой точки p до облака точек $O = \{p' \in \mathbb{R}^3\}$:

$$d(p, O) = \min_{p' \in O} \|p - p'\| \tag{2.5}$$

где операция $\|p - p'\|$ - задает Евклидово расстояние между точками p, p'.

Также введем h - пороговое значение расстояния, точки ближе которого считаются совпадающими и оператор Иверсона:

$$[x] = \begin{cases} 1, x \text{ is True} \\ 0, \text{otherwise} \end{cases} \tag{2.6}$$

Тогда для метрики соответствия облака точек S облаку точек G задаются следующим образом:

Precision (иногда «Accuracy»)

$$P(h) = \frac{100}{\|S\|} \sum_{p \in S} [d(p, G) < h] \tag{2.7}$$

Использовалась в работах [16, 24, 21].

Recall (иногда «Completition»)

$$R(h) = \frac{100}{\|G\|} \sum_{p \in G} [d(p, S) < h] \tag{2.8}$$

Использовалась в работах [24, 21].

F-score

$$F(h) = \frac{2 \cdot P(h) \cdot R(h)}{P(h) + R(h)} \tag{2.9}$$

Использовалась в работах [16, 30, 27, 37, 24, 22, 21].

Accuracy

$$A(h) = \frac{1}{\|S\|} \sum_{p \in S} d(p, G) \tag{2.10}$$

Использовалась в работе [42].

Completeness

$$C(h) = \frac{1}{\|G\|} \sum_{p \in G} d(p, S) \tag{2.11}$$

Использовалась в работе [42].

Chamfer distance

$$CD = \frac{1}{2|G|} \sum_{p \in S} (d(p, G))^2 + \frac{1}{2|S|} \sum_{p \in G} (d(p, S))^2 \tag{2.12}$$

Использовалась в работах [16, 30, 27, 37, 33, 19].

Chamfer L1 distance

$$CD = \frac{1}{2|G|} \sum_{p \in S} |d(p, G)| + \frac{1}{2|S|} \sum_{p \in G} |d(p, S)| \tag{2.13}$$

Использовалась в работе [42].

Hausdorff distance Прямое (расстояние Хаусдорфа):

$$HD^{forward} = \max_{p \in S} d(p, G) < h \tag{2.14}$$

Обратное (расстояние Хаусдорфа):

$$HD^{backward} = \max_{p \in G} d(p, S) < h \tag{2.15}$$

Использовалась в работе [23].

Normal consitency score Также для оценки качества поверхности может оцениваться соответствие нормалей к восстановленной поверхности с нормалями истинной поверхности. Пусть $N = (p', p'')|p' \in G, p'' = \min_{p \in S} d(p, G)$ - множество пар точек из G и S ближайших друг к другу, а $\vec{n'}, \vec{n''}$ - соответствующие им нормали, тогда метрика соответствия нормалей формулируется следующим образом:

$$NCS = \frac{1}{\|G\|} \sum_{p', p'' \in N} \|\vec{n'} \times \vec{n''}\| \tag{2.16}$$

Использовалась в работах [16, 30].

Также есть предложение [23] использовать метрики для оценки качества трехмерных карт используя в качестве истинных данных лидарные облака точек, но подход пока не является популярным.

2.5 Обзор современного состояния проблемы

Существующие подходы к реконструкции трехмерных карт

Существующие подходы к реконструкции трёхмерных карт можно условно разделить на 2 группы: аналитические и нейросетевые.

Аналитические методы основаны на использовании одной из математических моделей геометрической поверхности, описанных в п. 2.2 и последующей реконструкции меша при помощи алгоритма марширующих кубов 2.2. Примерами таких методов являются [19] - реконструкция карты при помощи SPSR и методы [35, 32] использующие TSDF. Преимуществом таких моделей является математическая строгость и часто скорость работы. При этом такие методы сильно восприимчивы к шуму, требовательны к памяти, при реконструкции детальных карт и ограничены используемой моделью реконструкции карты.

Также среди аналитических подходов набирает популярность методы использующие Гауссово рассеивание (Gaussian Splatting). На данный момент нет работ которые бы использовали этот метод для реконструкции меш-карт, но он используется для картирования [54, 47, 53] и есть подходы, исследующие возможность восстановления меша из таких карт [46, 39]. Также есть работы на пересечении неявных обучаемых представлений (раздел далее) и гауссова рассеивания [38].

Нейросетевые методы, в свою очередь делятся на 2 подгруппы: методы, использующие неявные обучаемые представления и методы, использующие явные обучаемые представления. В случае явных представлений, нейросети используются для обогащения входящих данных, например путём путем генерации вектора признаков для каждой точки [25] или выучивания некоторых универсальных паттернов, которые используются при реконструкции поверхности[41, 28]. При этом в реконструкции участвуют исходные облака точек. Преимуществом таких подходов является высокое качество реконструкции и универсальность, но такие методы требуют больших вычислительных ресурсов и зачастую плохо справляются с небольшими по площади объектами - дородными знаками, заборами.

В случае неявных представлений - входные данные используются для восстановления неявного представления поверхности, которое затем используется для реконструкции меша независимо от входных данных. Примерами таких методов являются [44, 43, 42, 49, 40, 52]. Преимуществом таких методов является их универсальность, способность работать с шумными данными, и эффективность представления в памяти и использования для дальнейших задач. Недостатками является также вычислительная емкость и качество реконструкции деталей.

Также можно выделить группу методов, реконструирующих карту с использованием семантической информации. Примерами таких методов являются [32, 42, 17, 43]. Недостаток этих подходов - использование семантической информации как дополнительного атрибута для хранения, но не как источника априорной информации о структуре сцены.

Архитектура современных подходов

Для иллюстрации характерных особенностей архитектуры современных подходов к реконструкции трехмерных карт, рассмотрим архитектуры аналитического метода VDB-Fusion[35] и нейросетевого подхода SHINE-Mapping[43].

VDB-Fusion [35] представляет карту в виде трёхмерной сетки, где каждая ячейка сетки хранит значение TSDF. Трехмерная решетка хранится в виде разреженной структуры данных VDB [5], которая представляет собой B-дерево фиксированной глубины (см. Рис. 9) и позволяет эффективно хранить в памяти трёхмерные данные, сохраняя быстрый доступ к ним. Каждая ячейка сетки хранит значение TSDF 2.2 и веса, который отвечает за уверенность в этом значении. Веса обновляются на основе весов предыдущего значения и значения нового измерения.

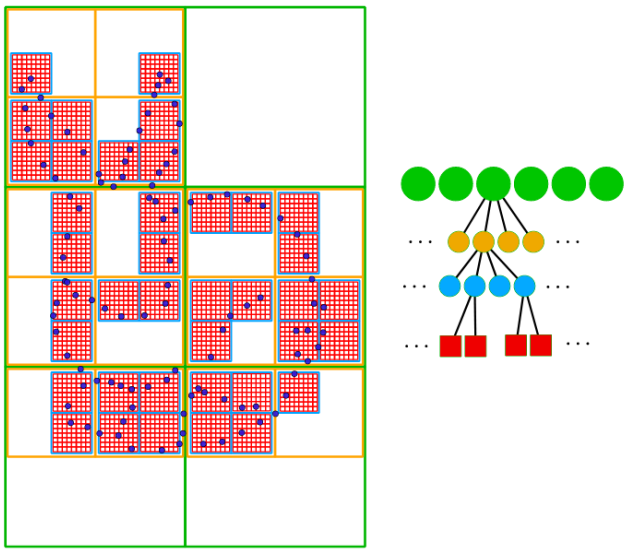


Рис. 9: Иллюстрация структуры данных VDB - B-дерево фиксированной глубины, оптимизированное для работы с 3D объектами.

Процедура картирования заключается в обновлении значений TSDF и весов в ячейках сетки на основе новых измерений. Для каждой точки измерения производится проекция на трёхмерную сетку и трассировка луча, соединяющего сенсор и поверхность, после чего обновляются значения TSDF и веса в ячейках, которые пересекает луч. Обновление производится на основе формулы:

$$\begin{aligned} D_t(x) &= \frac{D_{t-1}(x) \cdot W_{t-1}(x) + d_{t-1} \cdot w_{t-1}}{W_{t-1} + w_{t-1}} \\ W_t &= W_{t-1} + w_t \end{aligned} \tag{2.17}$$

где $D_t(x)$ - значение TSDF на момент времени t, W_t - вес на момент времени t, d_{t-1} - значение измерения на момент времени $t-1$, w_{t-1} - вес измерения на момент времени $t-1$. Вес измерения w_t вычисляется на основе расстояния до ближайшей ячейки сетки и углов падения лучей сенсора. Чем ближе измерение к ячейке, тем больше вес. Чем больше разность углов падения между измерениями в одной и той

же ячейке, тем меньше вес.

Пример интеграции точки в карту представлен на Рис. 10.

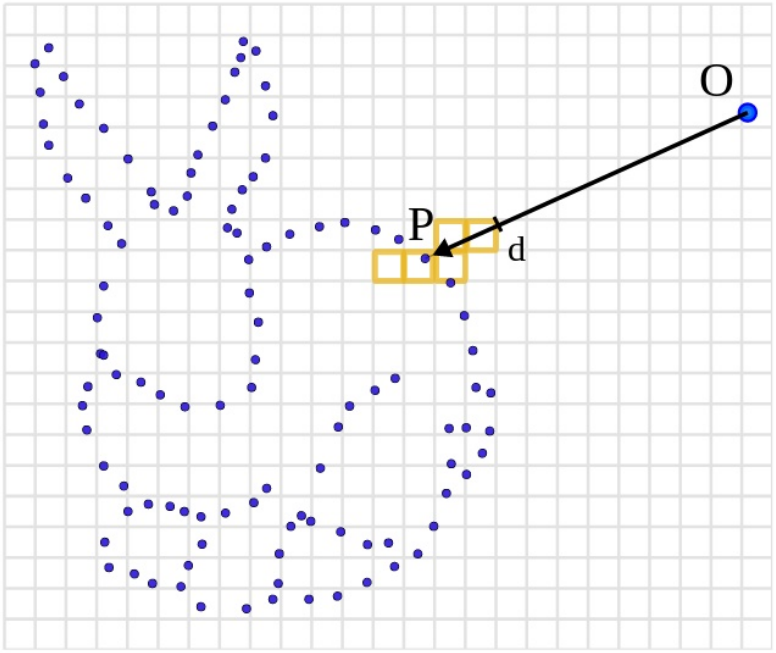


Рис. 10: Иллюстрация процесса интеграции новой точки на карту. Здесь O - сенсор, P - добавляемая точка, d - расстояние, в рамках которого обновляется карта (truncation distance). Ячейки обведённые жирным - получают обновление.

SHINE-Mapping [43] - нейросетевой алгоритм картирования на основе неявных обучаемых представлений. Он состоит из модуля предобработки данных, трёхмерной неявной карты и декодера. Общая схема работы алгоритма представлена на Рис. 11.

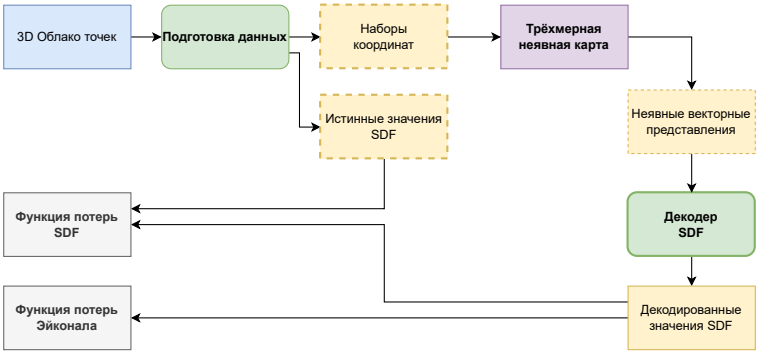


Рис. 11: Общая схема алгоритма SHINE-Mapping[43].

Подготовка данных состоит из следующих этапов:

1. Предобработка лидарных данных: удаление шумовых точек, фильтрация точек, приведение к общему формату.

2. Случайный отбор точек, которые будут использоваться для обучения системы.

3. Выборка точек вдоль лучей лидара и формирование обучающих пар точек и значений SDF в них.

Карта представлена в виде иерархической трёхмерной решётки, последние 3 слоя которой хранят в узлах векторные обучаемые представления. В этих представлениях закодировано поведение SDF функции в некоторой окрестности данного узла.

Пусть дана трёхмерная иерархическая решётка с уровнями детализации L^i, $i \in \{1, \ldots, N\}$, где N - количество уровней детализации. Пусть C_x^i - ячейка решётки, в которой содержится точка X на уровне L^i. Пусть X^i - локальные координаты точки X внутри ячейки C^i. Вершины ячейки (узлы решётки) C_x^i обозначаются как V_k^i, где $k \in \{1, \ldots, 8\}$, их координаты как V_{k,C^i}, векторные представления как $F_{V_k^i}$. D_{geo} и D_{sem} - декодеры геометрической и семантической информации соответственно.

Тогда векторное представление F_X^i точки X на уровне L^i может быть вычислено при помощи трилинейной интерполяции в соответствии с алгоритмом 2.1. Схематичное изображение процесса получения векторного представления точки X на уровне L^i изображено на Рис. 12.

Алгоритм 2.1 Вычисление SDF с использованием неявного обучаемого представления в алгоритме SHINE-Mapping[43].

Вход: L^i, $i \in \{1, \ldots, N\}$, X

Выход: $\text{SDF}(X)$, $\text{Label}(X)$

1: $X \to C_x^1, C_x^2, C_x^3$ — Проекция на решётку

2: **для** $i = 1 \to 3$

3: $\quad X^i = \left(X - V_{0,C^i}\right) / \left(V_{7,C^i} - V_{0,C^i}\right)$ — Локальные координаты внутри C_x^i[1]

4: $\quad F_X^i = \text{TrilinearInterpolation}(\{F_{V_k^i}\}_{k=0}^8, X^i)$ — Трилинейная интерполяция[2]

5: $F_X = \sum_{i=1}^3 F_X^i$ — Агрегирование признаков по слоям

6: $\text{SDF}(X) = D_{\text{geo}}(R_X)$ — Декодирование неявного представления

7: $\text{Label}(X) = D_{\text{sem}}(R_X)$

[1] V_{0,C^i} и V_{7,C^i} - координаты противоположных углов ячейки C_x^i

[2] Расчёт значения в точке X по значениям в узлах ячейки C_x^i

---

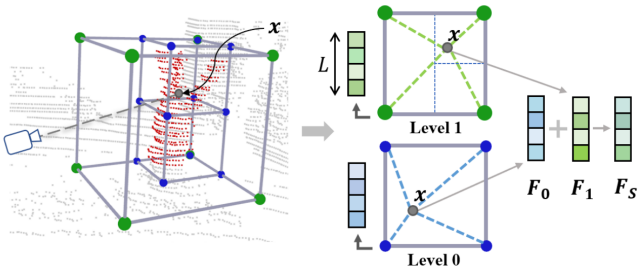


Рис. 12: Иллюстрация получения неявного векторного представления для конкретной точки в алгоритме SHINE-Mapping[43].

Декодеры геометрической и семантической информации D_{geo} и D_{sem} преобразуют векторное представление в значение SDF и семантическую метку соответственно. Каждый из них представляет собой многослойный перцептрон.

Обучение - реконструкция карты может производиться в двух режимах: итеративно (онлайн) и всей карты целиком (офлайн). При этом в процессе обучения происходит обновление векторных представлений узлов решётки, которые затем используются для реконструкции карты. И одновременно с этим обучается декодер, который преобразует векторное представление в значение SDF и семантическую метку.

При обучении используются две функции потерь:

- Функция потерь для SDF на основе бинарной кросс-энтропии:

$$L_{\text{SDF}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\sigma(d_i)\log(\sigma(\bar{d}_i)) + (1-\sigma(d_i)\log(1-\sigma(\bar{d}_i))\right) \tag{2.18}$$

- Функция потерь на основе функции Эйконала, для придания мешу гладкости:

$$L_{\text{eik}} = \frac{1}{N}\sum_{i=1}^{N}\left(\|\nabla f_\theta(\mathbf{x}_i)\|_2 - 1\right)^2 \tag{2.19}$$

Также может быть использован дополнительный декодер для семантической информации, в таком случае вместе с SDF алгоритм восстанавливает также семантическую метку в точке, а при обучении используется дополнительный член в функции - функция потерь для семантической информации представленная уравнением 2.20 также на основе бинарной кросс-энтропии.

$$L_{\text{sem}} = -\frac{1}{N}\sum_{i=1}^{N}\left(l_i\log(\hat{l}_i) + (1-l_i)\log(1-\hat{l}_i)\right) \tag{2.20}$$

Анализ существующих подходов

На данный момент аналитические методы показывают хорошие результаты и сильно обгоняют нейросетевые методы по скорости работы. Они требуют меньше вычислительных ресурсов и могут работать в real-time режиме. Однако они имеют ряд ограничений и недостатков: низкая устойчивость к шуму, необходимость хранить результаты работы в явном виде, что приводит к большим затратам памяти при реконструкции детализированных карт.

Метод гауссова рассеивания быстро набирает популярность и является очень перспективным для многих задач беспилотного транспорта, но также ограничен используемой математической моделью. Этот подход позволяет интегрировать семантическую информацию в карту, но только как дополнительный атрибут, а не как источник априорной информации о структуре сцены.

В свою очередь нейросетевые методы, в частности методы, которые восстанавливают карту в виде неявных представлений решают многие недостатки аналитических подходов: они более устойчивы к шуму, могут работать с разреженными данными, требуют меньше памяти и могут без дополнительных вычислений и преобразований использоваться в других нейросетевых алгоритмах.

Особенно перспективными в этом ключе являются семантические карты, которые помимо геометрической информации содержат и семантическую. Они позволяют использовать карты для более сложных задач, таких как навигация, планирование и другие, а также улучшают качество карты и упрощают её анализ.

Ключевыми недостатками нейросетевых методов являются вычислительная сложность и скорость работы. Что усложняет их использование в реальных задачах.

С точки зрения реконструкции семантических карт, большим пробелом в исследованиях является отсутствие алгоритмов, которые бы учитывали семантику при реконструкции карты. Существующие методы либо реконструируют карту без учёта семантики[44, 49, 40], либо просто «фиксируют» семантику на уже восстановленной карте[43, 52, 42]. В то время как использование семантики потенциально может ускорить процесс реконструкции карты и повысить качество результирующей карты.

Недостатки в виде скорости работы и вычислительной сложности адресуются работами [52, 44], тогда как проблема реконструкции семантических карт без использования семантической информации остаётся открытой.

2.6 Постановка задачи и цель работы

Постановка задачи в общем виде Реконструкции карты в виде меша является частным случаем задачи реконструкции трехмерной поверхности, в которой в качестве математической модели представления выбрана полигональная сетка.

Задача реконструкции поверхности в общем виде представляет собой задачу нахождения истинных параметров некоторого объекта, опираясь на множество наблюдений, где в качестве искомого объекта выступает непрерывная поверхность S^*, а наблюдениями является неупорядоченное трехмерное облако точек $P = \{p \in \mathbb{R}^3\}$ полученное при помощи некоторого сенсора глубины (например лидар).

При этом, с формальной точки зрения некорректно говорить о задаче реконструкции непрерывной поверхности S^* по дискретному облаку точек P, поэтому ставится задача восстановления «геометрической» аппроксимации S такой поверхности. И представляет собой следующую задачу оптимизации [34]:

$$\min_S L(S, P) + \lambda R(S) \tag{2.21}$$

где L - функция потерь отвечающая за соответствие восстановленной поверхности S и облака точек P, R - функция регуляризации, которая означает за «геометричность» восстановленной поверхности и λ - коэффициент регуляризации.

Замечание 2.1. Данная формулировка является чистой абстракцией ввиду сложности явной реализации описанных функций S и R. На практике эти функции реализуются косвенно - через другие задачи оптимизации, выбор которых осуществляется исследователем исходя из конкретной задачи и выбранного подхода.

Реконструкция поверхности в виде меша В такой постановке задачи, накладывается ограничение на математическое представление реконструируемой поверхности: поверхность состоит из набора вершин V и набора многоугольников F (граней), обычно треугольных:

$$S =< V, F > \tag{2.22}$$

$$V = \{P_i | P_i \in \mathbb{R}^3, 0 \leqslant i \leqslant N-1\} \tag{2.23}$$

$$F = \{(p, q, r, l)_j | 0 \leqslant p, q, r \leqslant N-1, 0 \leqslant j \leqslant K-1, l \in L\} \tag{2.24}$$

здесь:

- F - полигоны из которых состоит меш (как правило треугольники);
- p, q, r - номера вершин из множества V, из которых состоит многоугольник;
- l - семантическая метка из множества L семантических классов;
- N, K - количество вершин и многоугольников, из которых состоит поверхность.

В такой формулировке задачу оптимизации 2.21 можно уточнить, записав:

$$L(S, P) = \frac{1}{N_P} \sum_{p \in P} \min_{f \in F} d(f, p) - \lambda \sum_{f \in F} \sum_{l=1}^{L} l_f \log p_f \tag{2.25}$$

Здесь

- первое слагаемое - геометрическая функция потерь, где $d(f, p) \in \mathbb{R}$ - функция расстояния от точки p до участка поверхности f:
- второе слагаемое - семантическая функция потерь, где l_f - семантическая метка участка поверхности f, p_f - предсказанная алгоритмом вероятность метки l_f.

3 Семантически обусловленные алгоритмы реконструкции трёхмерной меш-карты по размеченным лидарным данным

В данном разделе рассмотрены два разработанных подхода к решению задачи семантически обусловленного картирования. Первый подход основан на использовании явных представлений (TSDF), второй подход основан на использовании неявных обучаемых представлений.

В разделе 3.1 рассмотрен алгоритм реконструкции трёхмерной семантической карты на основе явных представлений (TSDF) с использованием алгоритма VDB-Fusion[35]. В разделе 3.2 рассмотрен метод реконструкции трёхмерной семантической карты на основе неявных обучаемых представлений - его архитектура, функции потерь, процедура обучения и картирования.

3.1 Архитектура системы реконструкии трёхмерных семантических карт с использованием явных представлений (TSDF)

Система реконструкции трёхмерной семантической карты на основе явных представлений (TSDF) реализована на основе алгоритма VDB-Fusion[35] (см. раздел 2.5), с модификацией для учёта семантической информации.

Общая архитектура системы

Общая архитектура системы приведена на Рис. 13. Система состоит из N модулей картирования VDB-Fusion[35], каждый из которых отвечает за реконструкцию одного семантического слоя. Слои реконструируются независимо и затем объединяются в одну общую карту.

Представление карты в явном виде (TSDF)

Принцип работы модуля VDB-Fusion описан в разделе 2.5. Предложенная семантическая версия алгоритма VDB-Fusion-S, представлена на Рис. 13 (б). Отличие от несемантической версии заключается в модуле разделения облаков точек на се-

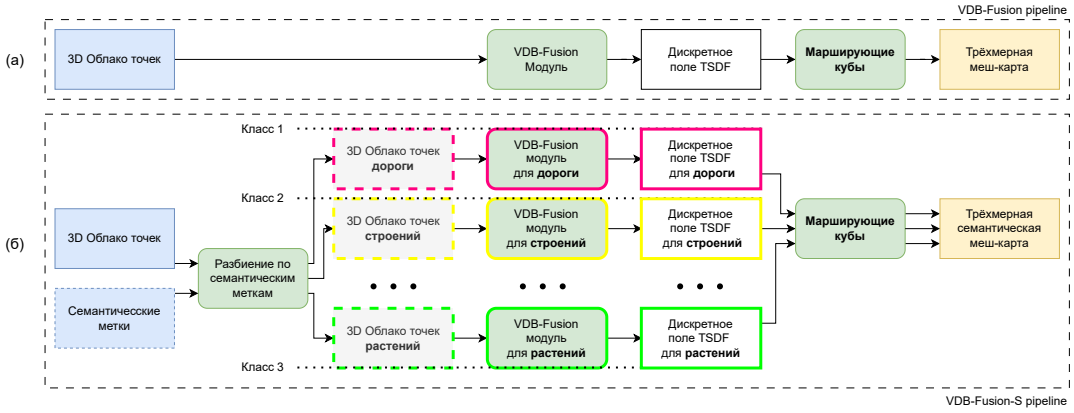


Рис. 13: Архитектура разработанной системы реконструкции трёхмерных карт на основе явных представлений (TSDF): VDB-Fusion[35], не семантическая (а), и предложенная архитектура VDB-Fusion-S, семантическая(б)

мантические слои, разделение на которое позволяет впоследствии восстановить семантическую карту.

Реконструкция семантической меш-карты

Реконструкция семантической карты происходит также при помощи алгоритма марширующих кубов (см. раздел 2.2), каждый семантический слой восстанавливается по отдельности, после чего слои объединяются в одну общую карту, где каждая вершина меша хранит семантическую метку соответствующую номеру слоя.

Мотивация

Новизна данного реализации заключается в архитектуре, учитывающей семантическую информацию. Предложенный алгоритм позволяет восстановить трёхмерную семантическую карту в явном виде при помощи аналитического алгоритма и сравнить с результатами полученными нейросетевым подходом.

3.2 Архитектура системы реконструкии трёхмерных семантических карт с использованием неявных обучаемых представлений

Система реконструкции трёхмерной семантической карты на основе неявных обучаемых представлений реализована на основе алгоритма SHINE-Mapping [43], подробно описанного в разделе 2.5. В данном разделе будут рассмотрена мотивация

выбора данного метода, архитектура системы, функции потерь, процедура обучения и картирования.

Общая архитектура системы

Архитектура разработанной системы приведена на рис Рис. 14. Система состоит из следующих модулей: модуль подготовки данных, трёхмерная неявная карта, в которой хранятся обучаемые представления; модуль семантической обусловленности; декодер геометрической информации; декодер семантической информации. Также на схеме отражены функции потерь используемые при обучении данной системы.

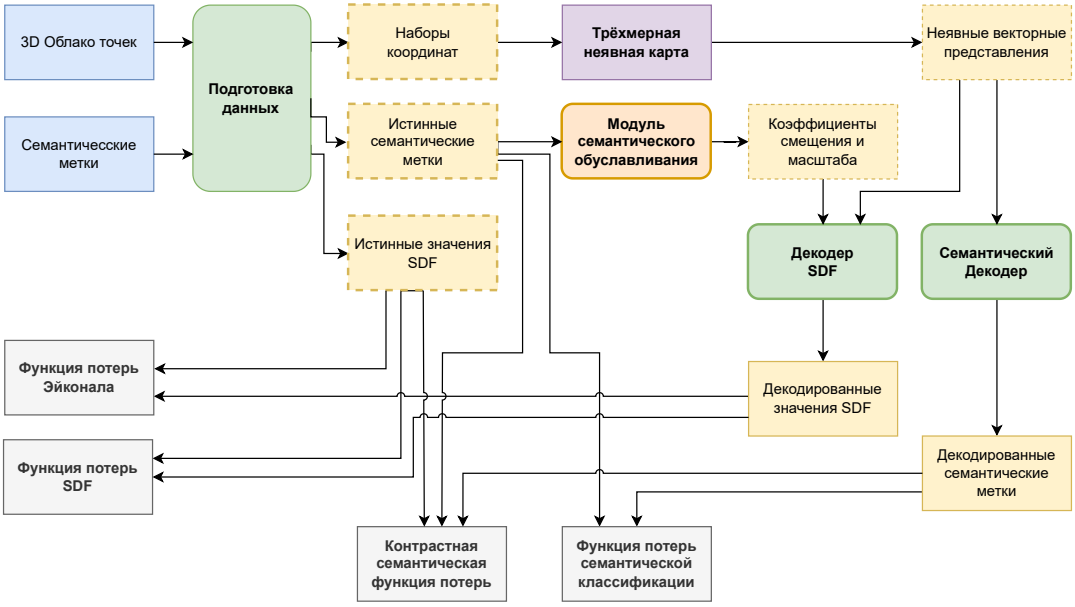


Рис. 14: Архитектура предложенной семантически обусловленной системы реконструкции трёхмерных карт на основе неявных обучаемых представлений.

Модуль подготовки данных

Подготовка данных включает те же шаги, что и для алгоритма SHINE-Mapping[43] описанные в разделе 2.5, но отличается подходом к отбору точек участвующих в картировании.

Предложенный метод подготовки данных включает следующие шаги:

- Выборка обучающих данных с учетом семантической разметки - точки, принадлежащие разным классам семантической разметки выбираются с разной

вероятностью, учитывающей неравномерность распределения классов в данных.

- Поиск и отбор точек принадлежащих «контурам» объектов - дополнительное отбираются точки, принадлежащие границам объектов, как наиболее информативные для реконструкции карты. Точки отбираются в соответствии с алгоритмом 3.1. Иллюстрация процесса отбора точек приведена на Рис. 15.

Алгоритм 3.1 Алгоритм получения контуров

Вход: Облако точек P, семантическая разметка точек L

Вход: Угловые разрешения лидара $\Delta\theta$, $\Delta\varphi$, количество точек в облаке N

Выход: Контуры R в исходном облаке точек

1: $\theta_{\max} = \max(\theta)$, $\varphi_{\max} = \max(\varphi)$[1]
2: $W = \theta_{\max}/\Delta\theta$, $H = \varphi_{\max}/\Delta\varphi$[2]
3: $I = \text{np.zeros}(W, H)$ — Создание пустого изображения
4: $\text{Mask} = \text{np.zeros}(N)$ — Создание маски с индексами точек
5: **для** $p_i \in P$
6: $p_i \to C_i$ — Проекция точки на сферу
7: $C_i \to \theta_i, \varphi_i$ — Преобразование в угловые координаты [3]
8: $\theta_i, \varphi_i \to w_i, h_i$ — Преобразование в координаты изображения
9: $I[w_i, h_i] = L_i$ — Проекция семантической разметки на изображение
10: $\text{Mask}[i] = \theta_i, \varphi_i$ — Сохранение индексов точек
11: $G = \text{cv2.Sobel}(I, \text{cv2.CV_16S}, 1, 0, ksize = 3)$[4,5]
12: $T = \text{cv2.adaptiveThreshold}(G, 0, 255, \dots)$[6]
13: $C = \text{cv2.findContours}(T, \dots)$[7]
14: **для** $c_i \in C$
15: **для** $p_j \in c_i$
16: $R \leftarrow \text{Mask}[p_j]$
17: **return** R

[1] Максимальные углы обзора лидара [2] Разрешение изображения [3] Шаг может быть опущен, если используется современный лидар, который предоставляет такую информацию
[4] Вычисление градиента изображения [5] Здесь и далее cv2 - библиотека OpenCV [6] Бинаризуем изображение при помощи адаптивного порога [7] Поиск контуров

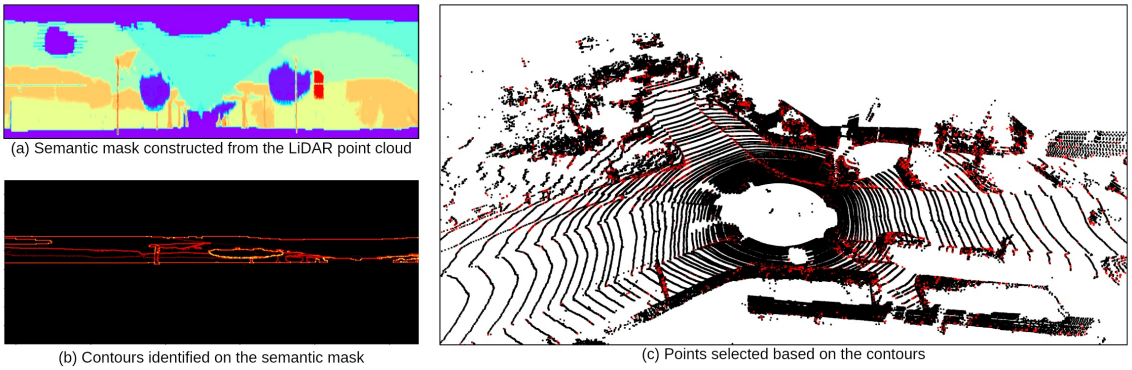


Рис. 15: Иллюстрация этапов разработанного алгоритма поиска контуров объектов на лидарном облаке точек.

Неявное обучаемое представление карты

В предложенном подходе преставление карты совпадает с представлением в алгоритме SHINE-Mapping[43] и описано в разделе 2.5.

Модуль семантической обусловленности и декодеры неявных преставлений

Модуль семантической обусловленности отвечает за обучение неявных представлений с учётом семантической информации. Он представляет собой многослойный перцептрон, который принимает на вход семантическую метку соответствующую точке и на выходе формирует коэффициенты модуляции признаков последнего слоя декодера неявных представлений.

Принцип работы модуля семантической обусловленности изображён на Рис. 16.

Этот подход был впервые предложен в работе [6] и позволяет просто и эффективно обуславливать работу декодера.

Архитектуры геометрического и семантического декодеров соответствуют архитектурам декодеров в алгоритме SHINE-Mapping[43] и описаны в разделе 2.5.

Процедуры обучения системы и картирования

Процесс обучения предложенной системы схож с процессом обучения алгоритма SHINE-Mapping[43]. Декодеры и неявное представление карты обучаются параллельно опираясь на функции потерь, описанные в разделе 2.5.

Процесс реконструкии карты отличается от такового в методе SHINE-Mapping[43], тем, что декодер геометрической информации обусловлен на семантику и требует перестроения архитектуры таким образом, чтобы декодеры работали последовательно

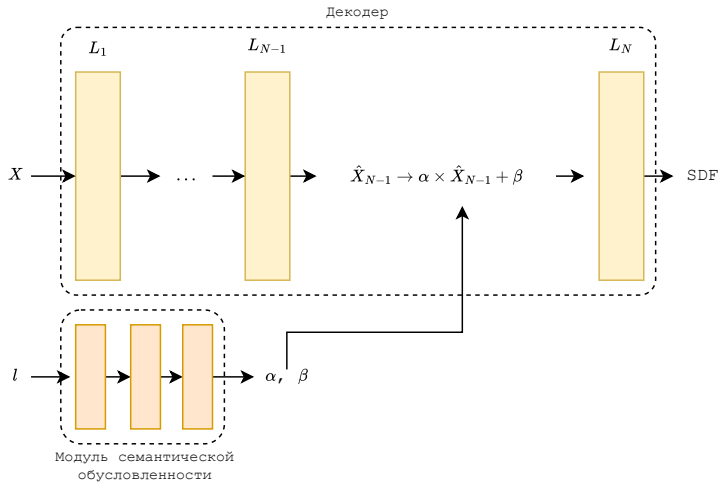


Рис. 16: Принцип работы модуля семантической обусловленности. Здесь $L_i(\cdot)$ - слой i многослойного перцептрона (декодера), X - векторное представление точки, $\hat{X}_i$ - векторное представление точки после слоя i декодера, l - семантическая метка точки, N - количество слоёв декодера.

и декодированная семантическая метка применялась при восстановлении геометрической информации. Отличие процесса реконструкции отражено на Рис. 17.

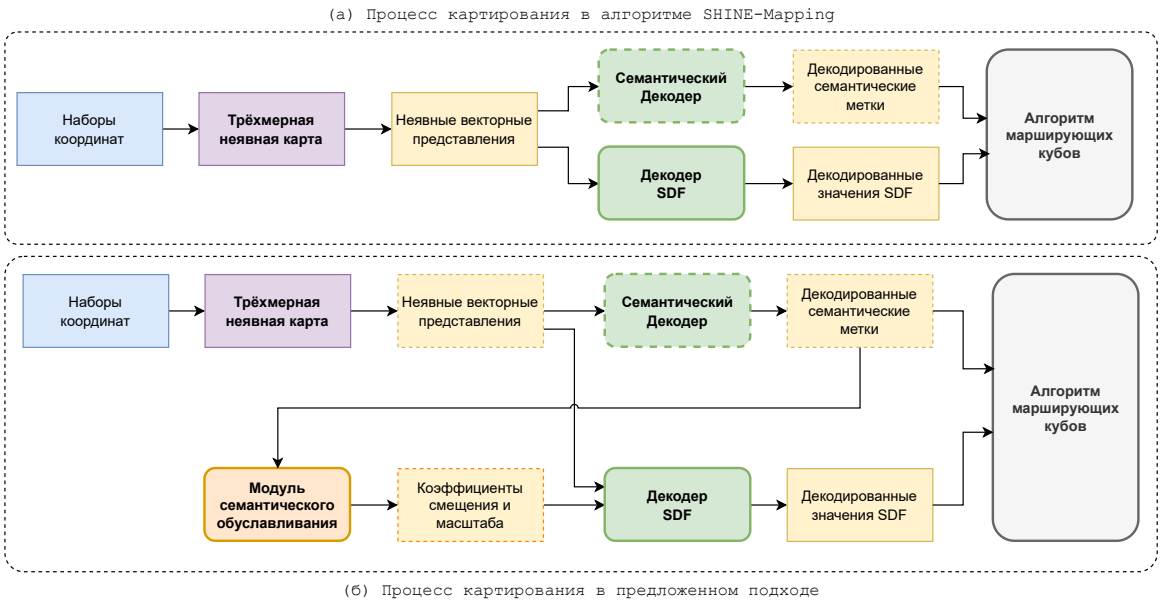


Рис. 17: Сравнительная схема подходов к картированию: (а) алгоритма SHINE-Mapping[43] и (б) предложенного подхода.

3.3 Функции потерь

Для обучения системы реконструкции трёхмерной семантической карты на основе неявных обучаемых представлений были использованы следующие функции потерь:

- **Функция потерь SDF** - L_{SDF} 2.18 - функция потерь, вычисляющая разницу между предсказанным значением SDF и истинным значением SDF в точке.
- **Функция потерь Эйконала** - L_{eik} 2.19 - функция потерь, вычисляющая разницу между предсказанным значением градиента SDF и истинным значением градиента SDF в точке.
- **Семантическая функция потерь** - L_{sem} 2.20 - функция потерь, вычисляющая разницу между предсказанной семантической меткой и истинной семантической меткой в точке.
- **Контрастная семантическая функция потерь** - L_{cont} 3.2 - функция потерь, вычисляющая разницу между предсказанным значением SDF и истинным значением SDF в точке.

Функции потерь SDF, Эйконала и семантическая функция потерь совпадают с функциями потерь в алгоритме SHINE-Mapping[43] и описаны в разделе 2.5. Мною была предложена и использована контрастная семантическая функция потерь, описанная далее.

Общий вид функции потерь, использовавшейся при обучении системы, приведён на (3.1).

$$L = \lambda_{\text{SDF}} \cdot L_{\text{SDF}} + \lambda_{\text{Eik}} \cdot L_{\text{Eik}} + \lambda_{\text{sem}} \cdot L_{\text{sem}} + \lambda_{\text{cont}} \cdot L_{\text{cont}} \tag{3.1}$$

здесь λ_{SDF}, λ_{Eik}, λ_{sem}, λ_{cont} - коэффициенты, определяющие вклад каждой функции потерь.

Контрастная семантическая функция потерь

Семантическая обусловленность предложенного подхода улучшает качество реконструкции карты, но при этом снижает информативность отдельных точек с точки зрения геометрической информации. Наблюдаемая точка дает представление о характере функции SDF в данной окрестности для конкретного семантического класса,

и не дает никакой информации о геометрии соседних объектов. Контрастная семантическая функция потерь адресует эту проблему, давая дополнительную информацию о геометрии объектов другого класса в окрестности заданной точки.

Контрастная семантическая функция потерь вычисляется по формуле (3.2). Иллюстрация процесса вычисления контрастной семантической функции потерь приведена на Рис. 18.

$$L_{\text{cont}} = \sum_{l'}^{C} L_{\text{BCE}} \left(\left| \bar{d}_l - \bar{d}_{l'} \right| - M \right) \tag{3.2}$$

где $d(\mathbf{x})$ - истинное значение SDF $\bar{d}(\mathbf{x})$ - восстановленное значение SDF, l_i- Истинная метка класса, C- Количество семантических меток, $\hat{l}$ - восстановленная метка, $\hat{p}_{i,c}$ - восстановленная вероятность метки c, M - «отступ» - минимальное расстояние между соседними классами.

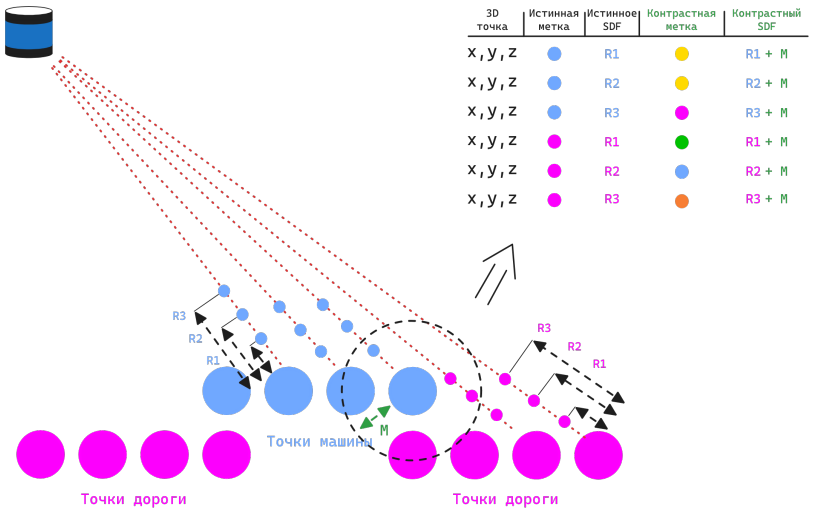


Рис. 18: Иллюстрация процесса вычисления предложенной контрастной семантической функции потерь, описанной в уравнении 3.2.

3.4 Оценка качества семантической меш-карты

Существующие метрики оценки трёхмерных семантических карт позволяют эффективно оценивать качество восстановленной трёхмерной поверхности, при наличии истинной трёхмерной поверхности. Однако, в случае масштабных трехмерных карт местности истинная трехмерная поверхность недоступна, и оценка качества трехмерной семантической карты становится проблематичной.

Кроме того, существующие подходы не учитывают семантику сцены, в то время

как реконструкция семантически размеченной карты позволяет оценивать качество реконструкции отдельных классов объектов и получать более детальное представление о достоинствах и недостатках алгоритма.

В ходе данной работы был разработан, предложен и опубликован подход к оценке качества трёхмерных семантических карт - SAMM[23]. Подход основан на вычислении метрик качества реконструкции семантической карты, таких как точность, полнота, F1-мера, а также метрик качества реконструкции геометрической карты, таких как Chamfer Distance и Hausdorff Distance - в отдельности для каждого семантического класса. Кроме того метод предлагает подход к оценке качества карт в отсутствие истинной трехмерной поверхности.

Математическое описание метрик качества реконструкции трехмерной семантической карты приведено в разделе 2.4. Сам подход описан в работе [23]. Предложенный подход использовался для оценки качества реконструкции трехмерных семантических карт в данной работе.

Новизна

Новизна предложенного подхода заключается в:

- Разработке и исследовании семантически обусловленного подхода к реконструкции трёхмерной карты на основе как явного аналитического подхода, так и неявных обучаемых представлений
- Предложении контрастной семантической функции потерь, учитывающей геометрическую информацию о соседних объектах
- Предложении метода подготовки данных, учитывающего семантическую информацию
- Разработке подхода к оценке качества семантической меш-карты

3.5 Детали реализации

Алгоритм VDB-Fusion-S реализован на языке Python. В основе лежит библиотека VDB-Fusion[35], реализованная на языке C++ и интерфейсом на языке Python. Для сегментации облаков точек и реконструкции меша использовались библиотеки:

- Open3D [8] - библиотека для работы с 3D сценами и объектами, включая меши и облака точек
- NumPy [15] - библиотека для быстрой работы с многомерными массивами на языке Python
- scikit-learn [2] - библиотека для машинного обучения на языке Python, в частности в ней реализованы KD-деревья, использованные при сегментации облаков точек

Алгоритм Sea-SHINE (семантически обусловленный алгоритм реконструкции трехмерных карт на основе неявных обучаемых представлений) реализован на языке Python. Он реализован на базе SHINE-Mapping[43], в основе которого лежат библиотеки:

- PyTorch [11] - библиотека для машинного обучения на языке Python, в частности в ней реализованы функции потерь, оптимизаторы и декодеры и автоматическое дифференцирование.
- kaolin [9] - библиотека - расширение для PyTorch компании Nvidia, предназначенная, быстрой для работы с трехмерными данными с использованием видеокарт. В алгоритме SHINE-Mapping с ее помощью реализовано октодерево, в котором хранятся неявные обучаемые представления.

Также в имплементации использовались библиотеки: Open3D[8], NumPy[15], OpenCV[50], scikit-learn[2].

Библиотека SAMM - библиотека для оценки качества трехмерных семантических карт, реализованная в рамках данной работы использует такие библиотеки как:

- Trimesh[45] - библиотека для работы с трехмерными мешами и облаками точек, в данной работе использовалась для работы с трехмерными мешами
- point-cloud-utils[36] - библиотека в которой реализовано большое количество функций для работы с облаками точек, в данной работе использовалась для выборки точек на поверхности меша и расчёте промежуточных величин при расчёте метрик

- LMDB[55] - библиотека для работы с базами данных, в данной работе использовалась для хранения и быстрого доступа к промежуточным результатам при расчёте метрик

Также использовались библиотеки: NumPy[15], scikit-learn[2].

4 Результаты тестирования разработанного метода картирования

Для оценки качества работы разработанных метода и алгоритма картирования проведены эксперименты на следующих наборах данных: Semantic KITTI, KITTI-360 и ITLP Campus. Наборы данных SemanticKITTI и KITTI-360 являются стандартными наборами данных для задачи семантической сегментации и картирования окружения. Набор данных ITLP Campus был собран и размечен в рамках данной работы.

В ходе экспериментов были исследованы следующие аспекты работы системы: эффект семантического обуславливания, эффект использования контрастной функции потерь, эффект семантически обусловленной выборки обучающих примеров. Также проведено сравнение результатов работы разработанной системы с существующими методами.

4.1 Описание используемых наборов данных

Semantic KITTI

Основным набором данных для оценки качества алгоритма картирования является SemanticKITTI[13]. Этот набор данных, предназначен для задач семантической сегментации 3D точечных облаков. Он является расширением популярного набора данных KITTI[3] и включает в себя аннотации для более чем 23 000 кадров (22 траектории).

Сбор данных для датасета осуществлялся с использованием автомобиля Volkswagen Passat B6, оснащенного различными сенсорами, установленными на крыше (см. Рис. 19). Основные сенсоры включают:

- Лидар Velodyne HDL-64E
- 2 стереокамеры Point Grey Flea 2 - цветные (FL2-14S3C-C) и монохромные (FL2-14S3M-C)
- GPS/IMU OXTS RT 3003

Эти сенсоры обеспечивают высокоточное трехмерное сканирование и позиционирование автомобиля, что позволяет создавать детализированные 3D модели окру-

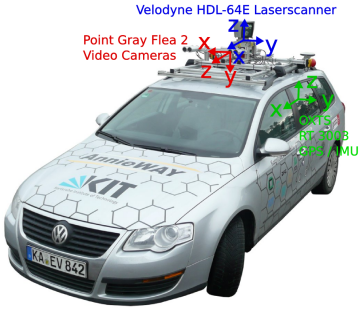


Рис. 19: Автомобиль использовавшийся для сбора набора данных SemanticKITTI[13], с установленными сенсорами

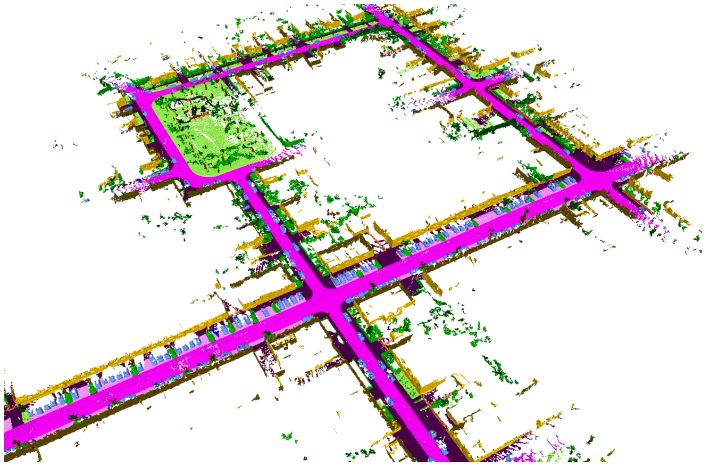


Рис. 20: Карта трека 00 из набора данных Semantic KITTI[13], построенная разработанным алгоритмом

жающего пространства. Наличие семантической разметки позволяет использовать этот набор данных для обучения и оценки подходов картирования.

В частности набор содержит:

- Общее количество кадров: более 23 000
- Количество семантических классов: 25
- Общая длина треков: около 39 км
- Трек 00: длина трека составляет 3.7 км и включает более 4 500 кадров

На Рис. 20 изображена семантическая карта трека 00 набора данных Semantic KITTI[13], построенная при помощи разработанного алгоритма.

Данный набор данных представляет интерес, так как является стандартом для оценки качества работы алгоритмов, в области беспилотного транспорта и картирования окружения.

KITTI-360

Набор данных KITTI-360[29] представляет собой своего рода расширение популярного набора данных KITTI[3], предназначенное для задач 3D картирования и семантической сегментации. Он включает в себя высокоточные аннотации для 360-градусных панорамных изображений и точечных облаков, собранных в разнообразных условиях.

Наборы данных KITTI-360 и Semantic KITTI не имеют общих траекторий, но для создания KITTI-360 использовался тот же автомобиль и частично те же сенсоры, что и для Semantic KITTI. Ключевые различия между KITTI-360 и Semantic KITTI заключаются в использовании более широкого набора сенсоров, разнообразной разметки и большем количестве данных.

Предметно, набор данных KITTI-360[29] содержит:

- Общее количество кадров: более 100 000
- Количество семантических классов: 19
- Общая длина треков: около 73 км
- Трек 00: длина трека составляет 8.8 км и включает более 10000 кадров

С точки зрения данной работы, основная ценность набора данных KITTI-360[29], в том, что он позволяет сравнить результаты работы разработанного алгоритма с результатами алгоритма NF-Atlas[42] - одного из современных методов также использующих семантику.

На Рис. 21 изображена семантическая карта трека 0 набора данных KITTI-360[29], построенная при помощи разработанного алгоритма.

ITLP Campus

Набор данных ITLP Campus был собран и размечен в рамках данной работы. Он представляет собой набор данных, собранный на территории кампуса МФТИ, и включает в себя аннотации для 3D облаков точек и изображений, собранных в различных условиях.

Набор собирался при помощи робототехнической платформы Husky A200, оснащенной различными сенсорами, установленными на специальной рейке (см. Рис. 22). Основные сенсоры включают:



Рис. 21: Карта трека 00 из набора данных KITTI-360[29], построенная разработанным алгоритмом

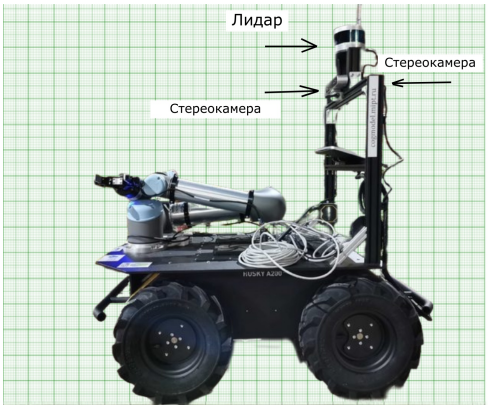


Рис. 22: Робототехническая платформа Husky, с помощью которой был собран набор данных ITLP Campus.

- Лидар Velodyne VLP-16
- Стереокамера ZED 2
- Сенсор инерции

Набор данных ITLP Campus включает 4 трека собранных по одному и тому же маршруту, но в разное время суток и в различных погодных условиях. Приблизительная протяженность каждого трека составляет 3,8 км. и включает порядка 4000 сканов (лидара и изображений).

Пример семантической карты из набора данных ITLP Campus показан на Рис. 24, пример данных из набора данных ITLP Campus показан на Рис. 23.

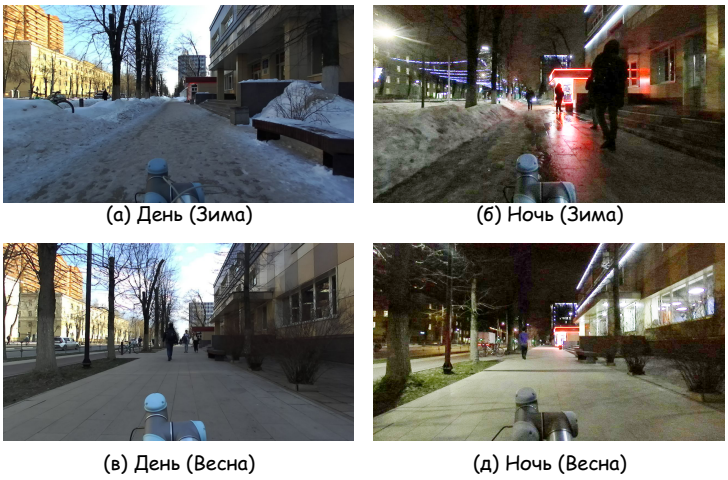


Рис. 23: Пример данных из набора данных ITLP Campus

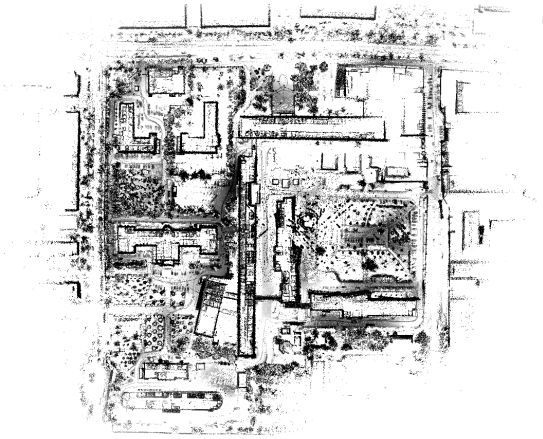


Рис. 24: Карта кампуса МФТИ, построенная по зимнему треку из набора данных ITLP Campus (вид сверху)

4.2 Подготовка данных

Наборы данных Semantic KITTI, KITTI-360 не нуждались в дополнительной подготовке, так как они уже содержат семантическую разметку. Для набора данных ITLP Campus была проведена проективная разметка семантических классов на изображениях и точечных облаках.

Разметка данных лидара

Для разметки данных лидара по семантическим маскам изображений был использован метод проективной разметки. Данный метод заключается в проецировании семантических масок изображений на точечное облако, полученное с лидара. Проекция производится при помощи калибровочных параметров камеры и лидара.

Алгоритм 4.1 Алгоритм темпоральной сегментации динамических объектов

Вход: Последовательность лидарных облаков точек P_t, окно N, радиус поиска d

Выход: Множество точек динамических объектов D_t

для $t = 1 \dots T$
  $P_{\text{window}} = \{P_{t-N/2}, \dots, P_t, \dots, P_{t+N/2}\}$ — Формирование окна кадров
  $D_t = \emptyset$ — Инициализация множества точек динамических объектов
  для $p_i \in P_t$
    $neighbors = 0$
    для $P_j \in P_{\text{window}}$
      если $P_j \neq P_t$ **то**
        $\text{tree} = \text{KDTree}(P_j)$ — Создание KD-дерева для облака P_j
        $\text{distances}, \text{indices} = \text{tree.query}(p_i, k = 1)$ — Поиск ближайших точек
        если $\text{distances} < d$ **то**
          $neighbors = neighbors + 1$
          break
    если $neighbors = 0$ **то**
      $D_t \leftarrow D_t \cup \{p_i\}$ — Присваиваем метку динамического объекта
return D_t

Для разметки изображений был использован нейросетевой алгоритм Oneformer [31], обученная на датасете Mapillary Vistas [7]. Этот набор данных содержит 66 семантических классов, которые были преобразованы в 19 классов, используемых в наборе данных SemanticKITTI для консистентности и удобства сравнения результатов.

Проективный подход к разметке лидарных данных схож с алгоритмом поиска и выделения контуров объектов 3.1, но проще и включает только прямую проекцию лидарного облака в плоскость изображений. Кроме того в дополнение к семантическим маскам изображений, был использован подход темпоральной сегментации динамических объектов, позволяющий удалять объекты не попавшие в зону видимости камеры Рис. 25. Подробнее подход описан в алгоритме 4.1.

Локализация робота

На используемой робототехнической платформе не установлен сенсор GNSS, поэтому для получения информации истинных поз робота использовались алгоритмы

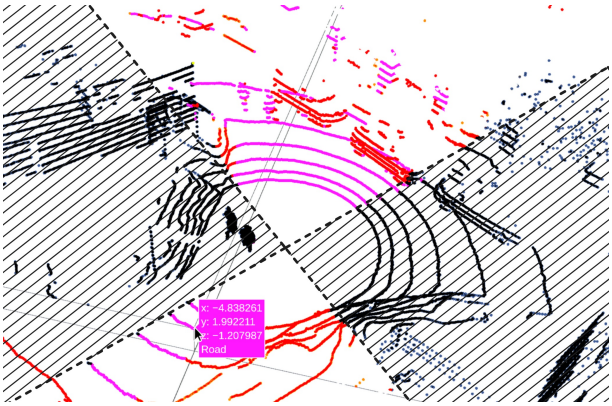


Рис. 25: Иллюстрация проективной сегментации лидарных данных по семантическим маскам изображений, заштрихованные области - не попадают в зону видимости камеры и остаются не размеченными.

одновременной локализации и построения карты (SLAM): ALeGO-LOAM[48] - алгоритм лидарной одометрии и InteractiveSLAM[18] для ручной коррекции карты.

4.3 Результаты экспериментов

В данном разделе приведены результаты экспериментов по оценке качества работы разработанной системы картирования на наборах данных Semantic KITTI, KITTI-360 и ITLP Campus. Для оценки качества работы системы использовались следующие метрики: точность (Precision), полнота (Recall), F-мера (F-score), расстояние Хаусдорфа (Hausdorff Distance) и расстояние Чамфера (Chamfer Distance) для различных семантических классов, методом предложенным в данной работе и описанным в работе [23], если не указано иного.

Для удобства сравнения предложенный алгоритм назван Sea-SHINE(сокращение от SEmantic Aware SHINE - семантически обусловленный SHINE). Промежуточные варианты алгоритма обозначаются как SHINE†, SHINE††, SHINE‡ и приведены в виде сносок в таблице. Лучшие результаты по каждой метрике выделены **жирным** шрифтом, вторые по значению - подчеркнуты. Из 20 классов датасета SemanticKITTI[13] здесь и далее для удобства восприятия приведены метрики для 4х из них: дорога, здания, дорожные знаки и растения.

В таблице 1 приведены результаты сравнения метрик качества реконструированных карт базовым решением SHINE-Mapping[43], предложенного алгоритма Sea-SHINE и промежуточных модификаций базового алгоритма с использованием предложенных подходов: семантической обусловленности, семантической выборки и кон-

трастной функции потерь, описанных в разделе 3.2. В первом разделе таблицы приведены общие метрики по всем семантическим классам, далее в секциях приведены метрики для отдельных семантических классов. Эксперименты проводились на 00 треке набора данных SemanticKITTI[13]. Параметры используемые в ходе эксперимента: количество итераций обучения - 40000, размер батча - 4096, отступ для контрастной функции потерь - 0.15, для реконструкции карты использовался каждый 10-й кадр. Коэффициенты суммирования функций потерь: $\lambda_{\text{SDF}} = 1.0$ - для функции потерь SDF, $\lambda_{\text{eik}} = \lambda_{\text{sem}} = \lambda_{\text{cont}} = 0.1$ - для остальных. Для реконструкции использовались кадры 0-2000.

Из таблицы 1 видно, что предложенный алгоритм Sea-SHINE показывает лучшие результаты по всем метрикам качества по сравнению с базовым решением SHINE-Mapping [43]. В частности, алгоритм Sea-SHINE показывает значимое улучшение для таких объектов, как дорожные знаки, которые являются важной частью карты в контексте беспилотного транспорта.

При этом можно заметить что предложенные улучшения не приводят к линейному увеличению всех метрик. Так для класса «дорога» лучшие результаты показывают промежуточные модификации алгоритма, а на втором месте по точности часто оказывается базовый алгоритм. Это связано с дизбалансом количества обучающей информации в процессе реконструкции карты. На класс дороги приходится значительный объем данных и при реконструкции карты без учета семантики, данный класс объектов измеряется чаще всего и точнее. Различие в метриках для промежуточных модификаций алгоритма и Sea-SHINE в свою очередь указывает на то, что предложенные модификации имеют конкурирующий эффект при обучении алгоритма, что приводит к улучшению качества в одних случаях и ухудшению в других. Таким образом несмотря на заметный прирост качества работы алгоритма, необходимо провести дополнительные исследования для оптимизации параметров обучения и архитектуры сети, в частично с учётом контекста применения алгоритма и наиболее значимых свойств карты окружения.

На рис. Рис. 26 приведены примеры реконструированных карт различными версиями алгоритма на треке 00 набора данных SemanticKITTI[13] и продемонстрировано качественное улучшение восстановленной карты, по сравнению с базовым решением. На карте восстановленной базовым алгоритмом SHINE-Mapping[43] видно, что отсутствует дорожный знак, в то время как на карте восстановленной алгоритмом

Таблица 1: Сравнение метрик качества реконструированных карт базовым решением SHINE-Mapping[43] и предложенного алгоритма Sea-SHINE на 00 треке набора данных SemanticKITTI[13]

Метод	$P \uparrow^{1}$	$R \uparrow^{2}$	$F \uparrow^{3}$	$HD_F \downarrow^{4}$	$HD_B \downarrow^{5}$	$CD \downarrow^{6}$
SHINE	82.96	76.208	78.28	56.427	82.09	3115.2
SHINE†	83.98	81.47	82.39	47.91	41.31	398.17
SHINE††	84.03	**81.56**	82.35	52.84	128.10	470.93
SHINE‡	**85.59**	81.08	**82.92**	49.79	64.73	385.065
Sea-SH*	84.90	81.49	82.84	**47.11**	**38.17**	**382.30**
Дорога:						
SHINE	99.23	98.61	**98.92**	1.41	13.33	0.0053
SHINE†	98.78	**98.73**	98.76	**0.48**	0.19	**0.00009**
SHINE††	97.57	97.56	97.56	4.08	0.22	0.0065
SHINE‡	**99.33**	98.15	98.73	0.61	0.24	0.00015
Sea-SH*	96.49	97.77	97.12	1.606	**0.16**	0.0009
Строения:						
SHINE	87.6	**97.32**	92.2	94.98	16.89	1.65
SHINE†	87.11	**97.32**	91.93	97.43	162.12	9.41
SHINE††	89.30	97.04	93.01	158.7	15.54	1.37
SHINE‡	**89.74**	96.82	**93.14**	101.5	**13.18**	1.98
Sea-SH*	89.44	96.88	93.01	**79.75**	37.09	**1.062**
Дорожные знаки:						
SHINE	76.95	25.32	38.1	321.853	0.15	35268.642
SHINE†	81.95	93.76	87.46	0.24	0.15	0.00055
SHINE††	69.40	95.10	80.24	**0.17**	0.22	0.001
SHINE‡	80.46	**95.60**	87.38	0.29	0.26	0.0019
Sea-SH*	**87.29**	92.96	**90.03**	0.38	**0.093**	**0.00044**
Растения:						
SHINE	84.59	85.30	84.94	53.94	313.58	782.72
SHINE†	84.92	**85.44**	85.18	**53.50**	**290.87**	600.20
SHINE††	88.33	84.35	**86.29**	57.10	297.42	506.45
SHINE‡	88.57	83.79	86.11	87.35	295.5	603.38
Sea-SH*	**90.24**	82.09	85.97	63.22	320.63	**489.14**

[1] Precision [2] Recall [3] F-score [4] Hausdorff Distance forward

[5] Hausdorff Distance backward [6] Chamfer Distance

† : SHINE + семантическая обусловленность †† : SHINE† + семантическая выборка ‡ : SHINE†† + контрастный лосс * : SHINE‡ + контуры объектов - предложенный алгоритм

Sea-SHINE дорожный знак восстановлен правильно.

4.4 Сравнение результатов с существующими методами

В данном разделе приведены результаты экспериментов по реконструкции карт другими современными алгоритмами и сравнение с результатами предложенно-

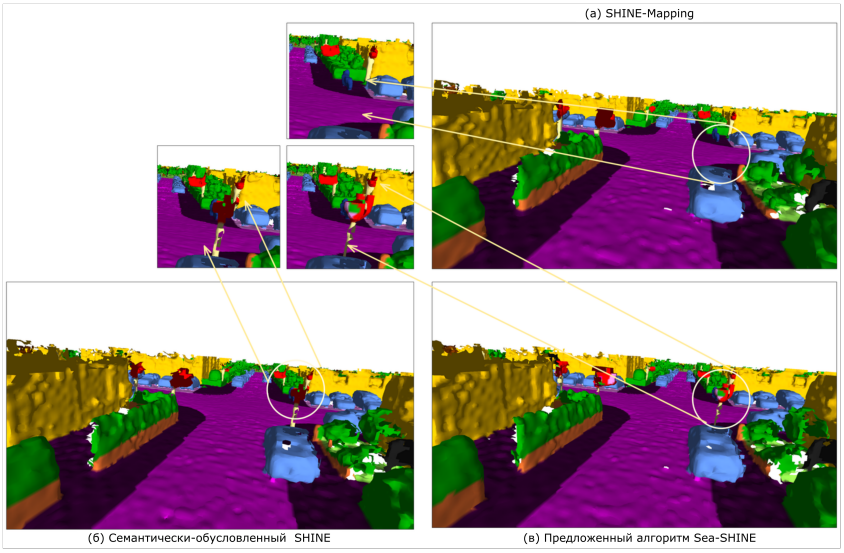


Рис. 26: Иллюстрация фрагмента карты трека 00 набора данных SemanticKITTI[13] восстановленная алгоритмами: (а) SHINE-Mapping[43], (б) семантичсеки-обсудловленной версией алгоритма SHINE и (в) предложенным алгоритмом Sea-SHINE.

Таблица 2: Ключевые параметры экспериментов с предложенным алгоритмом картирования, отличающиеся для различных наборов данных

Набор данных	N[1]	i[2]	**Первый кадр**[3]	**Последний кадр**[4]
S.-KITTI[5]	40000	10	0	2000
KITTI-360	5000	1	6330	6530
ITLP-Campus	40000	1	0	2893

[1] Количество итераций обучения [2] Для картирования использовался каждый i-шаг [3] Обучение началось на этом кадре [4] Обучение закончилось на этом кадре [5] SemanticKITTI [13]

го алгоритма Sea-SHINE. Эксперименты проводились на 00 треке набора данных SemanticKITTI[13], 00 треке набора данных KITTI-360[29] и зимнем треке набора данных ITLP Campus[51].

Параметры используемые в ходе эксперимента: размер батча - 4096, отступ для контрастной функции потерь - 0.15. Различающиеся параметры для каждого набора данных приведены в таблице 2.

Для оценки качества методов VDB-Fusion[35] и PIN-SLAM[52], так же как в разделе 4.3 использовались метрики: точность (Precision), полнота (Recall), F-мера (F-score), расстояние Хаусдорфа (Hausdorff Distance) и расстояние Чамфера (Chamfer Distance) для различных семантических классов в случае семантических карт и общие - в случае не семантических. В случае семантических карт приведены метрики

для 4-х из 20 классов набора данных SemanticKITTI[13]: дорога, здания, дорожные знаки и растения.

Для сравнения с алгоритмом NF-Atlas[42] карты построенные предложенным алгоритмом были использованы метрики описанные в работе [42]: полнота (MAE Completeness), точность (MAE Accuracy), расстояние Чамфера (Chamfer-L1 distance), полнота(recall), точность(precision) и F-мера. Подробнее эти метрики описаны в разделе 2.4.

Сравнение реконструкции меш-карты различными алгоритмами на наборе данных SemanticKITTI

В таблице 3 приведены результаты сравнения метрик качества трёхмерных не семантических карт построенных алгоритмами: VDB-Fusion[35], PIN-SLAM[52] и предложенным алгоритмом Sea-SHINE на 00 треке набора данных SemanticKITTI[13]. Из таблицы видно, алгоритм VDB-Fusion сильно превосходит другие алгоритмы по точности (Precision) и обратному расстоянию Хаусдорфа, но заметно уступает в полноте (Recall) и связанным абсолютным метрикам (прямое расстояние Хаусдорфа и расстояние Чамфера) нейросетевым подходам.

Сравнение предложенного подхода и алгоритма PIN-SLAM[52] показывает, что предложенный алгоритм Sea-SHINE показывает сравнимые результаты на уровне современных подходов. В частности, алгоритм Sea-SHINE менее чем на процент отстаёт по метрикам Precision P, Recall R и F-score F от лучшего алгоритма PIN-SLAM и показывает лучшие результаты по метрикам расстояния Хаусдорфа и Чамфера. Таким образом, предложенный алгоритм Sea-SHINE позволил уменьшить общую ошибку реконструкции карты, практически не потеряв в точности и полноте.

В то же время, PIN-SLAM[52] использует модифицированную архитектуру картирования при помощи неявных представлений и интеграция предложенных модификаций в алгоритм PIN-SLAM[52] может привести к дополнительному улучшению качества реконструкции карты.

Сравнение реконструкции семантической меш-карты различными алгоритмами на наборе данных SemanticKITTI

В таблице 4 приведены результаты сравнения метрик качества трёхмерных семантических карт построенных алгоритмами: PIN-SLAM[52], VDB-Fusion[35] и предложенным алгоритмом Sea-SHINE на 00 треке набора данных SemanticKITTI[13]. Из таблицы видно, что алгоритм PIN-SLAM[52] показывает лучшие результаты по пол-

Таблица 3: Сравнение метрик качества трёхмерных не семантических карт построенных различными алгоритмами на 00 треке набора данных SemanticKITTI[13]

Метод	$P \uparrow^{1}$	$R \uparrow^{2}$	$F \uparrow^{3}$	$HD_F \downarrow^{4}$	$HD_B \downarrow^{5}$	$CD \downarrow^{6}$
VDB-Fusion	**99.81**	78.06	**87.27**	178.1	**0.27**	710.0
PIN-SLAM	84.98	**82.06**	83.19	47.22	131.94	407.92
Sea-Shine	84.90	81.49	82.84	**47.11**	38.17	**382.3**

[1] Precision [2] Recall [3] F-score [4] Hausdorff Distance forward
[5] Hausdorff Distance backward [6] Chamfer Distance

ноте реконструкции дорожных знаков, в то время как алгоритм Sea-SHINE восстанавливает их более точно. PIN-SLAM заметно лучше справляется с реконструкцией растений, а предложенный алгоритм превосходит его по качеству реконструкции строений.

Результаты экспериментов показывают, что превосходство явной реконструкции карты (VDB-Fusion) с точки зрения точности (Precision) реконструкции наблюдается в любых семантических классах, но для некоторых классов (например дорога) метод неявной реконструкции PIN-SLAM приближается к результатам VDB-Fusion. Это говорит о том, что разрыв по качеству не является фундаментальным и может быть сокращён.

Также из таблицы 4 видно, что несмотря на отставание от алгоритма PIN-SLAM[52] в таблице 3, для классов строений и дороги предложенный алгоритм Sea-SHINE показывает заметно лучшие результаты по полноте реконструкции и лучшие по точности для классов строений и зданий. В то же время полнота реконструкции класса дорожных знаков существенно выше у алгоритма PIN-SLAM[52]. Таким образом предложенный подход к реконструкции трёхмерных семантических карт имеет как преимущества, так и недостатки в сравнении с существующими методами. Это подтверждает эффективность предложенного подхода, но также указывает на необходимость дальнейшего исследования и улучшения метода.

Также показательным является визуальное сравнение трёхмерных карт, построенных указанными методами на Рис. 27 и Рис. 28.

Из Рис. 27 видно, что причиной низкой полноты (Recall) у алгоритма VDB-

Таблица 4: Сравнение метрик качества трёхмерных **семантических** карт построенных различными алгоритмами на 00 треке набора данных SemanticKITTI[13]

Метод	$P \uparrow^{1}$	$R \uparrow^{2}$	$F \uparrow^{3}$	$HD_F \downarrow^{4}$	$HD_B \downarrow^{5}$	$CD \downarrow^{6}$
Дорога:						
VDB-F.[1]	**100.0**	86.63	92.83	83.25	**0.02**	23.21
PIN[3]	99.78	93.32	96.44	5.47	0.09	0.06
Sea-SH[4]	96.49	**97.77**	**97.12**	**1.606**	0.16	**0.0009**
Строения:						
VDB-F.	**98.43**	94.78	**96.57**	260.96	**2.08**	8.86
PIN	84.16	96.29	89.82	77.79	230.56	2.08
Sea-SH	89.44	**96.88**	93.01	**79.75**	37.09	**1.062**
Дорожные знаки:						
VDB-F.	**100.0**	86.42	**92.71**	0.49	**0.0045**	0.0006
PIN	85.03	**97.01**	90.63	**0.12**	0.15	**0.0004**
Sea-SH	87.29	92.96	90.03	0.38	0.093	**0.0004**
Растения:						
VDB-F.	**99.68**	73.67	84.73	472.0	**0.30**	**9.91**
PIN	91.13	**87.83**	**89.45**	63.80	319.01	342.68
Sea-SH	90.24	82.09	85.97	**63.22**	320.63	489.14

[1] Precision [2] Recall [3] F-score [4] Hausdorff Distance forward

[5] Hausdorff Distance backward [6] Chamfer Distance

[1] VDB-Fusion [3] PIN-SLAM [4] Semantic Aware SHINE

Fusion-S отчасти является его неспособность восстановить карту при низкой плотности входящих данных. На дорожном полотне отчётливо видны пропуски в реконструкции карты, которые связаны с тем что при реконструкции использовалось каждое 10-е облако точек (а не все) и в некоторых участках траектории данные принимали специфичный характер. Точки были видны под острым углом с разных ракурсов, из-за чего алгоритм VDB-Fusion не смог сделать корректное предположение о геометрии поверхности в этих областях. В то время как алгоритмы использующие неявные представления (PIN-SLAM[52] и Sea-SHINE) показывают более стабильные результаты, что связано с неявным представлением карты и его способностью обобщать данные.

На Рис. 28 хорошо видна разница в реконструкции дорожных знаков: VDB-Fusion-S[35] восстановил достаточно аккуратно основание, но очень фрагментарно восстановил сам знак. PIN-SLAM[52] восстановил знаки более полно, но с некоторыми артефактами. Предложенный алгоритм Sea-SHINE восстановил знаки более

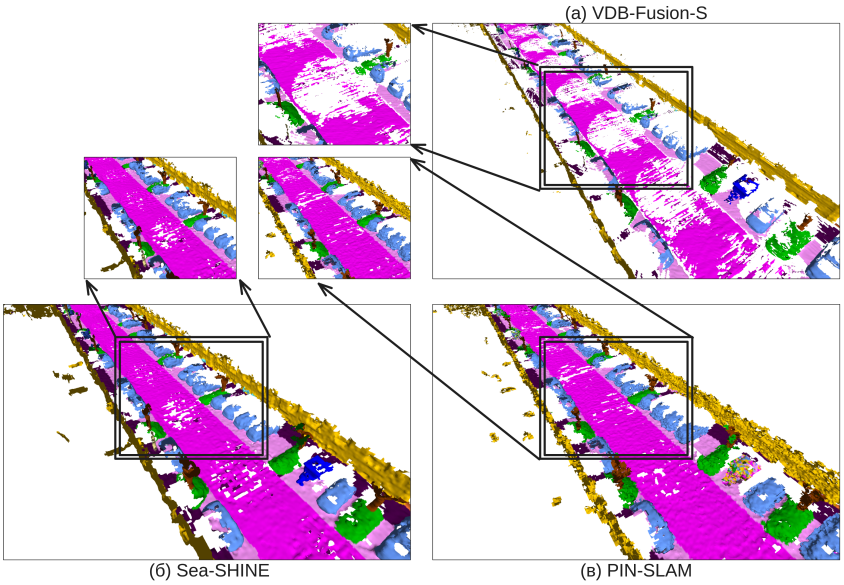


Рис. 27: Иллюстрация особенностей реконструкции дорожного полотна на примере карты трека 00 набора данных SemanticKITTI[13] восстановленная алгоритмами: (а) VDB-FUsion-S, (б) предложенным алгоритмом Sea-SHINE и (в) алгоритмом PIN-SLAM[52].

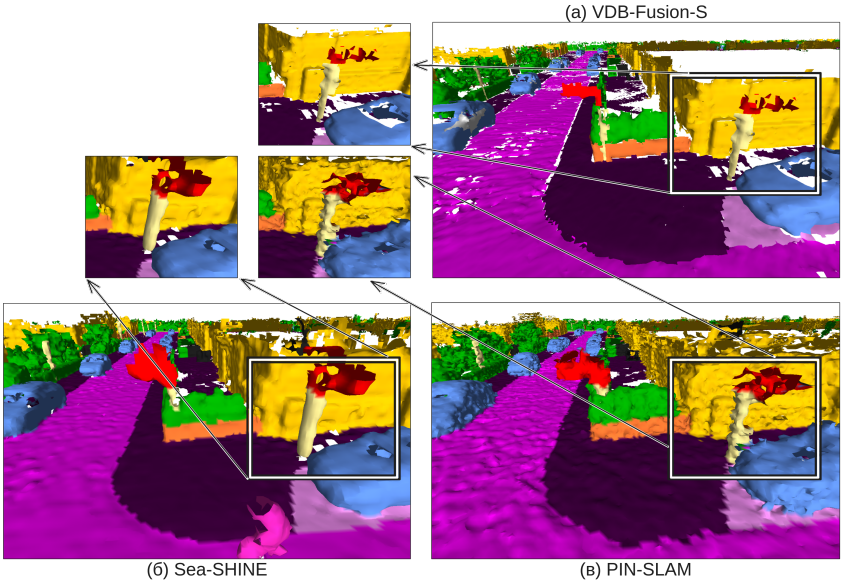


Рис. 28: Иллюстрация особенностей реконструкции дорожных знаков на примере карты набора данных SemanticKITTI[13] восстановленная алгоритмами: (а) VDB-FUsion-S, (б) предложенным алгоритмом Sea-SHINE и (в) алгоритмом PIN-SLAM[52].

аккуратно и почти без артефактов, что подтверждает результаты метрик качества.

Таблица 5: Сравнение метрик качества трёхмерных семантических карт, построенных различными алгоритмами на 00 треке набора данных KITTI-360[29]

Метод	$P \uparrow^{1}$	$R \uparrow^{2}$	$F \uparrow^{3}$	$Cmps \downarrow^{4}$	$Acc \downarrow^{5}$	$CD\text{-}L1 \downarrow^{6}$
NF-Atlas	94.85	**89.60**	**92.15**	**5.07**	3.79	**4.43**
SHINE[7]	**97.27**	84.20	90.26	7.20	**3.40**	5.30
Sea-SH[8]	95.50	87.93	91.56	6.33	4.29	5.30

[1] Precision [2] Recall [3] F-score [4] MAE Completeness

[5] MAE Accuracy [6] Chamfer Distance L1 [7] Базовый алгоритм SHINE-Mapping[43] [8] Sea-SHINE - предложенный алгоритм - семантически обусловленный SHINE

Сравнение реконструкции семантической меш-карты различными алгоритмами на наборе данных KITTI-360

В таблице 5 приведены результаты сравнения метрик качества трёхмерных семантических карт, построенных различными алгоритмами: NF-Atlas[42] и предложенным алгоритмом Sea-SHINE на 00 треке набора данных KITTI-360[29]. Метрики для алгоритма NF-Atlas[42] взяты из оригинальной публикации ввиду отсутствия возможности провести эксперименты с данным алгоритмом. Для расчёта метрик карты, восстановленной предложенным алгоритмом использовался код из репозитория авторов алгоритма NF-Atlas[42] и параметры описанные в публикации. Эксперимент проводился на 00 треке набора данных KITTI-360[29]. Для реконструкции карты использовался каждый кадр, меш-карта была построена с разрешением 10-см. Пороговое значение для расчёта метрик точности(Precision), полноты(Recall) и F-меры(F-score) составляло 0.1 метра.

Из результатов сравнения видно, что предложенный подход имеет неоднозначный эффект. Оригинальный алгоритм SHINE[43] показывает заметно лучшую точность (Precision, Accuracy) реконструкции, в сравнении с NF-Atlas[42], в то время как предложенная версия алгоритма уступает NF-Atlas по тем же параметрам. С другой стороны, предложенная версия алгоритма SHINE - заметно увеличивает показатели полноты (Recall, Completeness) восстановленной карты.

Таблица 6: Сравнение метрик качества трёхмерных не семантических карт построенных различными алгоритмами на зимнем треке набора данных ITLP Campus[51]

Метод	$P \uparrow^{1}$	$R \uparrow^{2}$	$F \uparrow^{3}$	$HD_F \downarrow^{4}$	$HD_B \downarrow^{5}$	$CD \downarrow^{6}$
VDB-FS.[7]	**56.70**	**45.51**	**49.54**	405.47	**132.14**	28971
Sea-SH[8]	45.54	44.75	44.11	**214.05**	262.39	**25403**
Дорога:						
VDB-FS.	**95.16**	**93.3**	**94.22**	16.09	362.54	**576.45**
Sea-SH	88.88	90.48	89.68	**6.93**	**361.92**	763.56
Строения:						
VDB-FS.	**70.78**	69.83	**70.31**	1024.83	149.73	1774.37
Sea-SH	57.50	**74.98**	65.09	**104.82**	**148.31**	**931.60**
Дорожные знаки:						
VDB-FS.	0.0	0.0	0.0	**27.40**	**27.40**	**750.72**
Sea-SH	**67.73**	**27.25**	**38.87**	402.71	192.35	1831.82
Растения:						
VDB-FS.	**80.39**	**82.39**	**81.38**	189.64	**439.31**	2316.098
Sea-SH	62.32	79.52	69.88	**90.34**	452.50	**3460.0**

[1] Precision [2] Recall [3] F-score [4] Hausdorff Distance forward

[5] Hausdorff Distance backward [6] Chamfer Distance [7] VDB-Fusion-S - модифицированный алгоритм [35], описанный в 3.1 [8] Предложенный алгоритм

Сравнение реконструкции семантической меш-карты различными алгоритмами на наборе данных ITLP-Campus

В таблице 6 приведены результаты сравнения метрик качества трёхмерных семантических карт, построенных алгоритмами: VDB-Fusion-S (описан в разделе 3.1) и предложенным алгоритмом Sea-SHINE на 00 треке набора данных ITLP-Campus[51]. Для оценки качества реконструированных карт использовались метрики: точность (Precision), полнота (Recall), F-мера (F-score), прямое расстояние Хаусдорфа (Hausdorff Distance forward) и обратное (Hausdorff Distance backward) и расстояние Чамфера (Chamfer Distance). Для предложенного алгоритма в таблице приведены 2 результата: базовая версия и версия использующая, неразмеченные данные для улучшения качества сематической карты.

Из таблицы 6 следует что метрики алгоритма VDB-Fusion-S превосходят метрики предложенного алгоритма Sea-SHINE, везде кроме дорожных знаков. Дорожные знаки алгоритм VDB-Fusion-S восстановил с метриками точности и полноты - 0. Это связано с особенностью набора данных ITLP-Campus[51]. Его разметка осу-

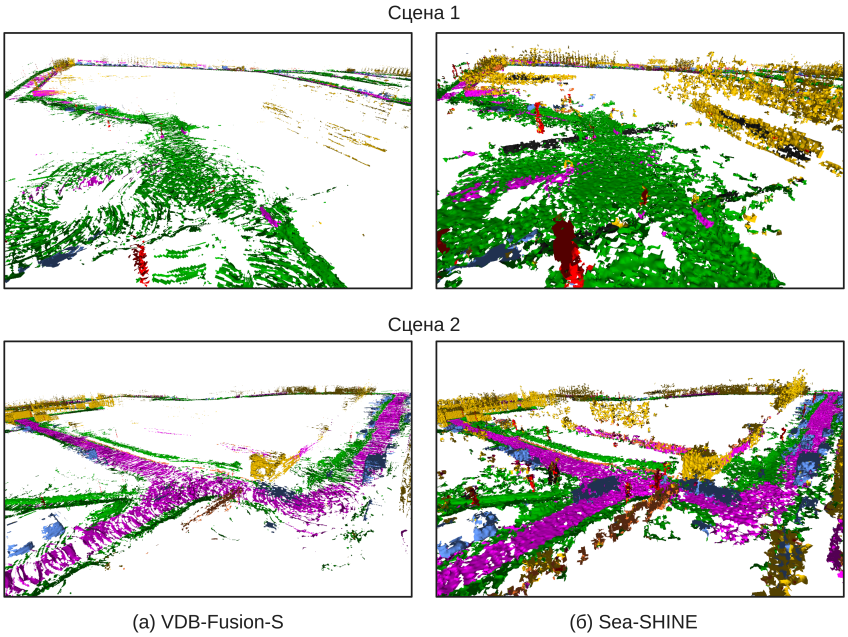


Рис. 29: Сравнение участков карт построенных алгоритмом VDB-Fusion-S и Sea-SHINE на наборе данных ITLP-Campus [51].

ществлялась проективным методом (см. раздел 4.1). В связи с этим, значимая часть лидарных данных не была размечена (см. Рис. 25) и не может использоваться для реконструкции карты. С учетом низкого разрешения лидара VLP-16 данные получаются очень разреженными (см. Рис. 29). В связи с этим когда речь идёт о достаточно протяженных объектах, таких как стены и здания - данных достаточно для восстановления модели, а при реконструкции дорожных знаков - нет.

Такая специфика набора данных накладывается на отсутствие истинной модели кампуса МФТИ, и оценка качества производится по лидарным данным - которых мало. Соответственно аналитический алгоритм VDB-Fusion-S, который просто фиксирует данные в явном формате получает высокие метрики там, где смог что-то восстановить. Предложенный алгоритм в свои очередь, восстанавливает карты из неявного представления и обобщает их. Это позволяет ему восстанавливать более плотную карту, но вместе с тем более шумную и менее точную. Это хорошо видно на Рис. 29.

4.5 Выводы

Эксперименты с базовым алгоритмом SHINE-Mapping[43] показали, что предложенные модификации алгоритма позволяют заметно повысить качество и преодолеть проблемы исходного алгоритма, связанные с реконструкцией небольших объектов,

мало представленных в данных. При этом сравнительные эксперименты эффективности предложенных модификаций по отдельности демонстрируют неоднозначность предложенных модификаций. Так, предложенные модификации алгоритма SHINE-Mapping[43] позволяют повысить качество, но не приводят к монотонному росту качества, реконструкции и требуют дополнительных исследований для оптимизации параметров обучения и архитектуры сети.

Эксперименты с другими современными алгоритмами реконструкции трёхмерных карт показали, что предложенный алгоритм Sea-SHINE показывает сравнимые результаты, но не показывает значимого прироста качества по сравнению с ними.

Эксперименты с разреженными данными показали, что предложенный подход к реконструкции карт показывает лучшие результаты по полноте восстановленной карты, но по точности сильно уступает классическому методу. Также эти эксперименты показали ограничения предложенного подход оценки качества трёхмерных карт.

Таким образом, из результатов экспериментов видно, что предложенная концепция семантической обусловленности при реконструкции трёхмерных карт методом неявных обучаемых представлений положительно влияет на качество реконструкции трёхмерных семантических карт, в частности небольших по протяженности объектов - таких, как дорожные знаки. Вместе с этим это влияет на качество реконструкции более крупных объектов, таких как здания и стены. Общее качество реконструкции карт не уступает, но и не превосходит существующие алгоритмы. Это говорит о том, что предложенный подход реконструкции карт с использованием семантической обусловленности имеет перспективы для применения в задачах беспилотного транспорта, но требует дополнительных исследований для оптимизации параметров обучения и архитектуры метода.

5 Заключение

В результате выполнения научно-исследовательской работы получены следующие результаты:

- Предложен подход семантически-обусловленной реконструкции трёхмерных семантических карт на основе неявных обучаемых представлений, а также реализован алгоритм Sea-SHINE, реализующий такой подход. Предложена семантическая контрастивная функция потерь и подходы к выборке обучающих примеров, учитывающие семантику сцены

- Разработан, реализован и опубликован подход к оценке качества трёхмерных семантических карт. Собран и размечен мультимодальный мультисезонный набор данных на базе кампуса МФТИ.

- Предложенный подход протестирован на публичных наборах данный и собственном, собранном в рамках данной работы. Результаты работы алгоритма оценены при помощи разработанного метода оценки качества, а также подходов описанных в современных работах по реконструкции трехмерных карт.

Полученные результаты показывают, что предложенный подход решает поставленную задачу реконструкции небольших элементов сцены. Разработанный алгоритм позволяет восстанавливать трёхмерные семантические карты по уровню реконструкции соответствующие современным работам, хотя и имеет некоторые ограничения и перспективу для улучшения.

Дальнейшие исследования предполагается посвятить повышению согласованности предложенных улучшений, а также ускорению работы предложенного алгоритма для работы в реальном времени.

Список литературы

1. Optimization and Learning for Rough Terrain Legged Locomotion / M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. Bagnell, C. Atkeson, J. Kuffner // I. J. Robotic Res. — 2011. — Янв. — Т. 30. — С. 175—191. — DOI: 10.1177/0278364910392608.

2. Scikit-Learn: Machine Learning in Python / F. Pedregosa [и др.] // Journal of Machine Learning Research. — 2011. — Т. 12, № 85. — С. 2825—2830. — ISSN 1533-7928. — URL: http://jmlr.org/papers/v12/pedregosa11a.html (дата обр. 16.06.2024).

3. *Geiger A.*, *Lenz P.*, *Urtasun R.* Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite // 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)). — Providence, RI : IEEE, 06/2012. — P. 3354–3361. — ISBN 978-1-4673-1228-8 978-1-4673-1226-4 978-1-4673-1227-1. — DOI: 10.1109/CVPR.2012.6248074. — URL: http://ieeexplore.ieee.org/document/6248074/ (visited on 06/07/2024).

4. *Kazhdan M.*, *Hoppe H.* Screened Poisson Surface Reconstruction // ACM Trans. Graph. — 2013. — June. — Vol. 32, no. 3. — P. 1–13. — ISSN 0730-0301, 1557-7368. — DOI: 10.1145/2487228.2487237. — URL: https://dl.acm.org/doi/10.1145/2487228.2487237 (visited on 12/17/2022).

5. OpenVDB: An Open-Source Data Structure and Toolkit for High-Resolution Volumes / K. Museth, J. Lait, J. Johanson, J. Budsberg, R. Henderson, M. Alden, P. Cucka, D. Hill, A. Pearce // ACM SIGGRAPH 2013 Courses. — New York, NY, USA : Association for Computing Machinery, 07.2013. — С. 1. — (SIGGRAPH '13). — ISBN 978-1-4503-2339-0. — DOI: 10.1145/2504435.2504454. — URL: https://doi.org/10.1145/2504435.2504454 (дата обр. 21.12.2022).

6. FiLM: Visual Reasoning with a General Conditioning Layer / E. Perez, F. Strub, H. de Vries, V. Dumoulin, A. Courville. — Вер. 1. — 22.09.2017. — DOI: 10.48550/arXiv.1709.07871. — arXiv: 1709.07871 [cs, stat]. — URL: http://arxiv.org/abs/1709.07871 (дата обр. 05.06.2024). — preprint.

7. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes / G. Neuhold, T. Ollmann, S. R. Bulo, P. Kontschieder // 2017 IEEE International Conference on Computer Vision (ICCV) (2017 IEEE International Conference on Computer Vision (ICCV)). — Venice : IEEE, 10/2017. — P. 5000–5009. — ISBN 978-1-5386-1032-9. — DOI: 10.1109/ICCV.2017.534. — URL: http://ieeexplore.ieee.org/document/8237796/ (visited on 06/07/2024).

8. *Zhou Q.-Y.*, *Park J.*, *Koltun V.* Open3D: A Modern Library for 3D Data Processing. — Вер. 1. — 29.01.2018. — DOI: 10.48550/arXiv.1801.09847. — arXiv: 1801.09847 [cs]. — URL: http://arxiv.org/abs/1801.09847 (дата обр. 16.06.2024). — preprint.

9. Kaolin: A PyTorch Library for Accelerating 3D Deep Learning Research / K. M. Jatavallabhula, E. Smith, J.-F. Lafleche, C. F. Tsang, A. Rozantsev, W. Chen, T. Xiang, R. Lebaredian, S. Fidler. — Вер. 1. — 12.11.2019. — DOI: 10.48550/arXiv.1911.05063. — arXiv: 1911.05063 [cs]. — URL: http://arxiv.org/abs/1911.05063 (дата обр. 16.06.2024). — preprint.

10. Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping / A. Rosinol, M. Abate, Y. Chang, L. Carlone. — Вер. 1. — 06.10.2019. — DOI: 10.48550/arXiv.1910.02490. — arXiv: 1910.02490 [cs]. — URL: http://arxiv.org/abs/1910.02490 (дата обр. 17.06.2024). — preprint.

11. PyTorch: An Imperative Style, High-Performance Deep Learning Library / A. Paszke [и др.]. — 03.12.2019. — DOI: 10.48550/arXiv.1912.01703. — arXiv: 1912.01703 [cs, stat]. — URL: http://arxiv.org/abs/1912.01703 (дата обр. 16.06.2024). — preprint.

12. Robust Object-based SLAM for High-speed Autonomous Navigation / K. Ok, K. Liu, K. Frey, J. P. How, N. Roy // 2019 International Conference on Robotics and Automation (ICRA). — 05.2019. — С. 669—675. — DOI: 10.1109/ICRA.2019.8794344.

13. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences / J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, J. Gall. — 16.08.2019. — arXiv: 1904.01416 [cs]. — URL: http://arxiv.org/abs/1904.01416 (дата обр. 23.04.2024). — preprint.

14. SuMa++: Efficient LiDAR-based Semantic SLAM / X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, C. Stachniss // 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). — 11.2019. — C. 4530—4537. — DOI: 10.1109/IROS40897.2019.8967704. — arXiv: 2105.11320 [cs]. — URL: http://arxiv.org/abs/2105.11320 (дата обр. 16.12.2022).

15. Array Programming with NumPy / C. R. Harris [et al.] // Nature. — 2020. — Sept. — Vol. 585, no. 7825. — P. 357–362. — ISSN 1476-4687. — DOI: 10.1038/s41586-020-2649-2. — URL: https://www.nature.com/articles/s41586-020-2649-2 (visited on 06/16/2024).

16. *Chen Z., Zhang H.* Neural Marching Cubes // ACM Trans. Graph. — 2021. — Дек. — Т. 40, № 6. — С. 1—15. — ISSN 0730-0301, 1557-7368. — DOI: 10.1145/3478513.3480518. — arXiv: 2106.11272 [cs]. — URL: http://arxiv.org/abs/2106.11272 (дата обр. 16.12.2022).

17. In-Place Scene Labelling and Understanding with Implicit Scene Representation / S. Zhi, T. Laidlow, S. Leutenegger, A. J. Davison. — arXiv, 08.2021. — DOI: 10.48550/arXiv.2103.15875. — arXiv: 2103.15875 [cs]. — URL: http://arxiv.org/abs/2103.15875 (дата обр. 22.12.2022).

18. Interactive 3D Graph SLAM for Map Correction / K. Koide, J. Miura, M. Yokozuka, S. Oishi, A. Banno // IEEE Robotics and Automation Letters. — 2021. — Jan. — Vol. 6, no. 1. — P. 40–47. — ISSN 2377-3766, 2377-3774. — DOI: 10.1109/LRA.2020.3028828. — URL: https://ieeexplore.ieee.org/document/9212613/ (visited on 06/07/2024).

19. Poisson Surface Reconstruction for LiDAR Odometry and Mapping / I. Vizzo, X. Chen, N. Chebrolu, J. Behley, C. Stachniss // 2021 IEEE International Conference on Robotics and Automation (ICRA). — Xi'an, China : IEEE, 05/2021. — P. 5624–5630. — ISBN 978-1-72819-077-8. — DOI: 10.1109/ICRA48506.2021.9562069. — URL: https://ieeexplore.ieee.org/document/9562069/ (visited on 12/10/2022).

20. SA-LOAM: Semantic-aided LiDAR SLAM with Loop Closure / L. Li, X. Kong, X. Zhao, W. Li, F. Wen, H. Zhang, Y. Liu. — arXiv, 07.2021. — arXiv: 2106.11516 [cs]. — URL: http://arxiv.org/abs/2106.11516 (дата обр. 16.12.2022).

21. *Song S.*, *Cui Z.*, *Qin R.* Vis2Mesh: Efficient Mesh Reconstruction from Unstructured Point Clouds of Large Scenes with Learned Virtual View Visibility. — arXiv, 08.2021. — DOI: 10.48550/arXiv.2108.08378. — arXiv: 2108.08378 [cs]. — URL: http://arxiv.org/abs/2108.08378 (дата обр. 16.12.2022).

22. *Ummenhofer B.*, *Koltun V.* Adaptive Surface Reconstruction with Multiscale Convolutional Kernels // 2021 IEEE/CVF International Conference on Computer Vision (ICCV). — Montreal, QC, Canada : IEEE, 10/2021. — P. 5631–5640. — ISBN 978-1-66542-812-5. — DOI: 10.1109/ICCV48922.2021.00560. — URL: https://ieeexplore.ieee.org/document/9711368/ (visited on 12/16/2022).

23. *Bezugly V.*, *Yudin D.* Reconstruction of 3D Semantic Map and Its Quality Estimation //. — 31.10.2022. — С. 325—335. — ISBN 978-3-031-19619-5. — DOI: 10.1007/978-3-031-19620-1_31.

24. BNV-Fusion: Dense 3D Reconstruction Using Bi-level Neural Volume Fusion / K. Li, Y. Tang, V. A. Prisacariu, P. H. Torr // 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — New Orleans, LA, USA : IEEE, 06/2022. — P. 6156–6165. — ISBN 978-1-66546-946-3. — DOI: 10.1109/CVPR52688.2022.00607. — URL: https://ieeexplore.ieee.org/document/9880298/ (visited on 12/16/2022).

25. *Boulch A.*, *Marlet R.* POCO: Point Convolution for Surface Reconstruction. — arXiv, 03.2022. — arXiv: 2201.01831 [cs]. — URL: http://arxiv.org/abs/2201.01831 (дата обр. 11.12.2022).

26. Dynamic Semantic Occupancy Mapping Using 3D Scene Flow and Closed-Form Bayesian Inference / A. Unnikrishnan, J. Wilson, L. Gan, A. Capodieci, P. Jayakumar, K. Barton, M. Ghaffari. — arXiv, 09.2022. — arXiv: 2108.03180 [cs]. — URL: http://arxiv.org/abs/2108.03180 (дата обр. 16.12.2022).

27. GIFS: Neural Implicit Function for General Shape Representation / J. Ye, Y. Chen, N. Wang, X. Wang. — arXiv, 07.2022. — arXiv: 2204.07126 [cs]. — URL: http://arxiv.org/abs/2204.07126 (дата обр. 11.12.2022).

28. *Huang J.*, *Chen H.-X.*, *Hu S.-M.* A Neural Galerkin Solver for Accurate Surface Reconstruction // ACM Transactions on Graphics. — 2022. — Dec. — Vol. 41, no. 6. — P. 1–16. — ISSN 0730-0301, 1557-7368. — DOI: 10.1145/3550454.

3555457. — URL: https://dl.acm.org/doi/10.1145/3550454.3555457 (visited on 03/14/2024).

29. *Liao Y.*, *Xie J.*, *Geiger A.* KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. — 06/03/2022. — arXiv: 2109.13410 [cs]. — URL: http://arxiv.org/abs/2109.13410 (visited on 06/07/2024). — preprint.

30. Neural Dual Contouring / Z. Chen, A. Tagliasacchi, T. Funkhouser, H. Zhang // ACM Trans. Graph. — 2022. — Июль. — Т. 41, № 4. — С. 1—13. — ISSN 0730-0301, 1557-7368. — DOI: 10.1145/3528223.3530108. — arXiv: 2202.01999 [cs]. — URL: http://arxiv.org/abs/2202.01999 (дата обр. 16.12.2022).

31. OneFormer: One Transformer to Rule Universal Image Segmentation / J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, H. Shi. — 26.12.2022. — DOI: 10.48550/arXiv.2211.06220. — arXiv: 2211.06220 [cs]. — URL: http://arxiv.org/abs/2211.06220 (дата обр. 07.06.2024). — preprint.

32. Panoptic Multi-TSDFs: A Flexible Representation for Online Multi-resolution Volumetric Mapping and Long-term Dynamic Scene Consistency / L. Schmid, J. Delmerico, J. Schönberger, J. Nieto, M. Pollefeys, R. Siegwart, C. Cadena // 2022 International Conference on Robotics and Automation (ICRA). — 05/2022. — P. 8018–8024. — DOI: 10.1109/ICRA46639.2022.9811877. — arXiv: 2109.10165 [cs]. — URL: http://arxiv.org/abs/2109.10165 (visited on 12/10/2022).

33. Probabilistic Implicit Scene Completion / D. Zhang, C. Choi, I. Park, Y. M. Kim. — arXiv, 04.2022. — arXiv: 2204.01264 [cs]. — URL: http://arxiv.org/abs/2204.01264 (дата обр. 16.12.2022).

34. Surface Reconstruction from Point Clouds: A Survey and a Benchmark / Z. Huang, Y. Wen, Z. Wang, J. Ren, K. Jia. — arXiv, 05.2022. — arXiv: 2205.02413 [cs]. — URL: http://arxiv.org/abs/2205.02413 (дата обр. 16.12.2022).

35. VDBFusion: Flexible and Efficient TSDF Integration of Range Sensor Data / I. Vizzo, T. Guadagnino, J. Behley, C. Stachniss // Sensors. — 2022. — Feb. — Vol. 22, no. 3. — P. 1296. — ISSN 1424-8220. — DOI: 10.3390/s22031296. — URL: https://www.mdpi.com/1424-8220/22/3/1296 (visited on 12/10/2022).

36. *Williams F.* Point Cloud Utils. — 2022. — URL: https://www.github.com/fwilliams/point-cloud-utils (дата обр. 16.06.2024).

37. *Xiao Y.*, *Xu J.*, *Gao S.* TaylorImNet for Fast 3D Shape Reconstruction Based on Implicit Surface Function. — arXiv, 01.2022. — arXiv: 2201.06845 [cs]. — URL: http://arxiv.org/abs/2201.06845 (дата обр. 16.12.2022).

38. *Chen H.*, *Li C.*, *Lee G. H.* NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance. — 12/01/2023. — arXiv: 2312.00846 [cs]. — URL: http://arxiv.org/abs/2312.00846 (visited on 06/16/2024). — preprint.

39. *Guédon A.*, *Lepetit V.* SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. — 02.12.2023. — arXiv: 2311.12775 [cs]. — URL: http://arxiv.org/abs/2311.12775 (дата обр. 16.06.2024). — preprint.

40. NeRF-LOAM: Neural Implicit Representation for Large-Scale Incremental LiDAR Odometry and Mapping / J. Deng, X. Chen, S. Xia, Z. Sun, G. Liu, W. Yu, L. Pei. — 19.03.2023. — arXiv: 2303.10709 [cs]. — URL: http://arxiv.org/abs/2303.10709 (дата обр. 21.01.2024). — preprint.

41. Neural Kernel Surface Reconstruction / J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, F. Williams. — 09.06.2023. — arXiv: 2305.19590 [cs]. — URL: http://arxiv.org/abs/2305.19590 (дата обр. 14.03.2024). — preprint.

42. NF-Atlas: Multi-Volume Neural Feature Fields for Large Scale LiDAR Mapping / X. Yu, Y. Liu, S. Mao, S. Zhou, R. Xiong, Y. Liao, Y. Wang. — 20.06.2023. — arXiv: 2304.04624 [cs]. — URL: http://arxiv.org/abs/2304.04624 (дата обр. 21.01.2024). — preprint.

43. SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations / X. Zhong, Y. Pan, J. Behley, C. Stachniss. — 20.02.2023. — DOI: 10.48550/arXiv.2210.02299. — arXiv: 2210.02299 [cs]. — URL: http://arxiv.org/abs/2210.02299 (дата обр. 22.05.2023). — preprint.

44. 3QFP: Efficient Neural Implicit Surface Reconstruction Using Tri-Quadtrees and Fourier Feature Positional Encoding / S. Sun, M. Mielle, A. J. Lilienthal, M. Magnusson. — 13.01.2024. — arXiv: 2401.07164 [cs]. — URL: http://arxiv.org/abs/2401.07164 (дата обр. 21.01.2024). — preprint.

45. *Dawson-Haggerty M.* Mikedh/Trimesh. — 16.06.2024. — URL: https://github.com/mikedh/trimesh (дата обр. 16.06.2024).

46. DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing / M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, J. Kannala. — 03/26/2024. — arXiv: 2403.17822 [cs]. — URL: http://arxiv.org/abs/2403.17822 (visited on 06/16/2024). — preprint.

47. DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes / X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, M.-H. Yang. — 20.03.2024. — DOI: 10.48550/arXiv.2312.07920. — arXiv: 2312.07920 [cs]. — URL: http://arxiv.org/abs/2312.07920 (дата обр. 23.04.2024). — preprint.

48. *jyakaranda*. Jyakaranda/A-LeGO-LOAM. — 27.04.2024. — URL: https://github.com/jyakaranda/A-LeGO-LOAM (дата обр. 07.06.2024).

49. N^{3}-Mapping: Normal Guided Neural Non-Projective Signed Distance Fields for Large-scale 3D Mapping / S. Song, J. Zhao, K. Huang, J. Lin, C. Ye, T. Feng. — 07.01.2024. — arXiv: 2401.03412 [cs]. — URL: http://arxiv.org/abs/2401.03412 (дата обр. 21.01.2024). — preprint.

50. Opencv/Opencv. — OpenCV, 16.06.2024. — URL: https://github.com/opencv/opencv (дата обр. 16.06.2024).

51. OPR-Project/ITLP-Campus. — OPR-Project, 12.04.2024. — URL: https://github.com/OPR-Project/ITLP-Campus (дата обр. 08.06.2024).

52. PIN-SLAM: LiDAR SLAM Using a Point-Based Implicit Neural Representation for Achieving Global Map Consistency / Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, C. Stachniss. — 17.01.2024. — arXiv: 2401.09101 [cs]. — URL: http://arxiv.org/abs/2401.09101 (дата обр. 21.01.2024). — preprint.

53. SGS-SLAM: Semantic Gaussian Splatting For Neural Dense SLAM / M. Li, S. Liu, H. Zhou, G. Zhu, N. Cheng, T. Deng, H. Wang. — 26.03.2024. — arXiv: 2402.03246 [cs]. — URL: http://arxiv.org/abs/2402.03246 (дата обр. 25.05.2024). — preprint.

54. Street Gaussians for Modeling Dynamic Urban Scenes / Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, S. Peng. — 02.01.2024. — DOI: 10.48550/arXiv.2401.01339. — arXiv: 2401.01339 [cs]. — URL: http://arxiv.org/abs/2401.01339 (дата обр. 23.04.2024). — preprint.

55. *Watson N.* Jnwatson/Py-Lmdb. — 15.06.2024. — URL: https://github.com/jnwatson/py-lmdb (дата обр. 16.06.2024).

56. *Kazhdan M.*, *Bolitho M.*, *Hoppe H.* Poisson Surface Reconstruction. —.

57. *Lorensen W. E.*, *Cline H. E.* Marching Cubes: A High Resolution 3D Surface Construction Algorithm. —.