|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт кибернетики

Кафедра проблем управления

**ОТЧЁТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ**

**Тема практики:** Разработка алгоритмов построения воксельной карты внешней среды автономного мобильного робота

приказ университета о направлении на практику

от «10» февраля 2020г. № 759-с

Отчет представлен к

рассмотрению:

Студент группы: КРМО-01-18 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Лесив

«\_\_» \_\_\_\_\_\_ 2020г.

Отчет утвержден.

Допущен к защите:

Руководитель практики

от кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.А.К. Диане

«\_\_» \_\_\_\_\_\_ 2020г.

Руководитель практики

от Университета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Н. Цыпкин

«\_\_» \_\_\_\_\_\_ 2020г.

Москва 2020

|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт кибернетики

Кафедра проблем управления

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРЕДДИПЛОМНУЮ ПРАКТИКУ**

**Студенту 2 курса учебной группы КРМО-01-18**

**Лесив Евгению Андреевичу**

**Место и время практики:** научная лаборатория кафедры проблем управления

**1. ЦЕЛЕВАЯ УСТАНОВКА:** Разработка алгоритмов построения воксельной карты внешней среды автономного мобильного робота

**2. СОДЕРЖАНИЕ ПРАКТИКИ:**

2.1 Изучить: подходы к представлению среды функционирования автономного мобильного робота.

2.2 Практически выполнить: разработать программное обеспечение для конвертации 3D облака точек в воксельное представление.

2.3 Ознакомиться: со средствами моделирования и визуализации сред функционирования автономных мобильных роботов.

**3.ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ:** -

**4. ОГРАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ:** -

Заведующий кафедрой:

«\_\_»\_\_\_\_\_\_\_\_\_\_2020г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.П. Романов

СОГЛАСОВАНО:

Руководитель практики от кафедры:

«\_\_»\_\_\_\_\_\_\_\_\_\_2020г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.А.К. Диане

Руководитель практики от Университета:

«\_\_»\_\_\_\_\_\_\_\_\_\_2020г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Сухоленцева

Задание получил:

«\_\_»\_\_\_\_\_\_\_\_\_\_2020г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Лесив

|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

**РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ ПРЕДДИПЛОМНОЙ ПРАКТИКИ**

студента Лесив Евгения Андреевича 2 курса группы КРМО-01-18 очной формы обучения, обучающегося по направлению подготовки 15.04.06, «Мехатроника и робототехника»

профиль «Интеллектуальные мобильные роботы»

|  |  |  |  |
| --- | --- | --- | --- |
| **Неделя** | **Сроки выполнения** | **Этап** | **Отметка о выполнении** |
| 3 | 10.02.20 -15.03.20 | Изучение подходов к представлению среды функционирования автономного мобильного робота |  |
| 4-6 | 16.03.20 - 14.04.20 | Ознакомление со средствами моделирования и визуализации сред функционирования автономных мобильных роботов |  |
| 7-9 | 15.04.20 –  05.05.20 | Разработка программного обеспечения для конвертации 3D облака точек в воксельное представление |  |
| 10-14 | 05.05.20- 23.05.20 | Проведение экспериментальных исследований |  |

**Согласовано:**

|  |  |  |
| --- | --- | --- |
| Заведующий кафедрой | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | М.П. Романов |
| Руководитель практики от кафедры | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | С.А.К. Диане |
|  |  |  |
| Руководитель практики от Университета | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.А. Сухоленцева |
| Обучающийся | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Е.А. Лесив |

**Проведенные инструктажи:**

Охрана труда: «\_\_»\_\_\_\_\_\_\_\_\_\_2020г.

Инструктирующий \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (С.А.К. Диане)

Инструктируемый \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Е.А. Лесив)

Техника безопасности: «\_\_»\_\_\_\_\_\_\_\_\_\_2020г.

Инструктирующий \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (С.А.К. Диане)

Инструктируемый \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Е.А. Лесив)

Пожарная безопасность: «\_\_»\_\_\_\_\_\_\_\_\_\_2020г.

Инструктирующий \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (С.А.К. Диане)

Инструктируемый \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Е.А. Лесив)

С правилами внутреннего распорядка ознакомлен: «\_\_»\_\_\_\_\_\_\_\_\_\_\_2020г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Е.А.Лесив)

**ОТЧЕТ**

**по преддипломной практике**

**студента 2 курса учебной группы КРМО-01-18**

**Института кибернетики**

**Лесив Евгения Андреевича**

1. Практику проходил с10.02.2020г. по 23.05.2020г.

В научно-исследовательской лаборатории кафедры проблем управления Российского технологического университета

1. Задание на практику выполнил в полном объеме

Не выполнены следующие задания: -

Подробное содержание выполненной на практике работы и достигнутые результаты: Изучил подходы к представлению среды функционирования автономного мобильного робота: программное обеспечение для конвертации 3D облака точек в воксельное представление; Ознакомился со средствами моделирования и визуализации сред функционирования автономных мобильных роботов;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Предложения по совершенствованию организации и прохождения практики: предложений нет

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_ \_(Е.А. Лесив)

(подпись)

«\_\_»\_\_\_\_\_\_\_\_\_\_2020г.

1. Заключение руководителя практики от профильной организации:

Студент приобрел следующие профессиональные навыки:

- способность разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования

- готовность участвовать в составлении аналитических обзоров и научно-технических отчетов по результатам выполненной работы, в подготовке публикаций по результатам исследований и разработок

Проявил себя как: организованный, целеустремленный специалист, способный самостоятельно решать поставленные перед ним инженерно-технические задачи и представить отчет по практике в полном объеме и в установленный срок и заслуживает оценки «отлично».

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель практики от кафедры проблем управления**

к.т.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.А.К. Диане

(подпись)

**Отчет проверил:**

**Руководитель практики от Университета**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Н. Цыпкин

Оглавление

[Введение 7](#_Toc41075181)

[1. Обзор программных средств моделирования и визуализации сред функционирования автономных мобильных роботов 7](#_Toc41075182)

[1. 1. Microsoft robotics developer studio 7](#_Toc41075183)

[1. 2. Dyn-soft robsim 5 8](#_Toc41075184)

[1. 3. Gazebo 9](#_Toc41075185)

[1. 4. Anykode marilou robotics studio 10](#_Toc41075186)

[1. 5. V-REP 11](#_Toc41075187)

[2. Анализ подходов к представлению среды функционирования автономного мобильного робота 12](#_Toc41075188)

[2. 1. OctoMap 12](#_Toc41075189)

[2. 2. GPOctoMap 14](#_Toc41075190)

[2. 3. BGKOctoMap 16](#_Toc41075191)

[3. Разработка алгоритма построения семантической воксельной карты внешней среды автономного мобильного робота 19](#_Toc41075192)

[Заключение 21](#_Toc41075193)

[Список использованных источников 22](#_Toc41075194)

[Приложение А: Пример файла запуска воксельной карты ROS Gazebo Rviz 23](#_Toc41075195)

# Введение

На сегодняшний день для решения задач робототехники большое значение имеют технологии интеллектуального анализа визуальной информации и картографирования, применение которых позволяет существенно повысить эффективность, надежность и автономность робототехнических систем.

Для функционирования робота в окружающей среде, помимо своего местоположения ему необходимо учитывать положение объектов реального мира. Одним из вариантов представления данных окружающего мира может служить семантическая карта. Возможность построения карты на борту робота позволяет выполнять ряд основополагающих задач автономной мобильной робототехники, таких как поиск и взаимодействие с объектами, планирование перемещений, координирование движений групп роботов. Таким образом, наличие высокоуровневой модели среды в формате семантической карты позволяет принять обоснованное решение на тактическом и стратегическом уровнях системы управления.

# 1. Обзор программных средств моделирования и визуализации сред функционирования автономных мобильных роботов

На сегодняшний день разработано большое количество программных средств моделирования робототехнических систем. Каждая среда обладает собственной спецификой, необходимой для решения различных прикладных задач. Таким образом, выбор среды моделирования зависит от разработки как самого робота, так и алгоритмов его взаимодействия с реальным миром.

## 1. 1. Microsoft robotics developer studio

Рассматриваемая среда объединяет в себе большое множество аппаратного обеспечения, обладает совместимостью с большим набором робототехнических платформ, среди которых: Boe-Bot, Aldebaran Robotics, Parallax Scribbler и еще множество других. Структура Microsoft Robotics Developer Studio включает в себя 4 основных модуля:

Первый модуль Visual Programming Language обладает возможностью описания алгоритмов в виде взаимосвязанных схем и диаграмм. Данный язык написания программного обеспечения разрабатывался для среды Microsoft Robotics Developer Studio с цель создания упрощенного метода реализации алгоритмов.

Второй модуль Concurrency and Coordination Runtime предполагает ведение разработки на основе многопоточного потока данных в синхронном и асинхронном режимах. Основу данного модуля составляет .NET Framework. Реализация данного модуля обусловлена потребностью одновременной обработки больших объемов данных с информационно-измерительной подсистемы роботов для мощных вычислительных устройств. Стоит отметить, что данная библиотека также используется и в других проектах, благодаря упрощенной реализации данного ряда задач.

Третий модуль Decentralized Software Services представляет упрощенную среду разработки, позволяющую реализовывать управление простыми компонентами роботов, влияющими на систему управления.

Четвертый модуль представляет собой среду для осуществления визуального моделирования процесса управления роботами.

Для осуществления симуляции физический процессов используется движок NVIDIA PhysX.

Разработка в MRDS ведется на явыках программирования JScript, Visual Basic, C# .NET и IronPython [1].

## 1. 2. Dyn-soft robsim 5

Данное программное обеспечение является разработкой компании Дин-Софт. RobSim 5 открывает возможности для осуществления моделирования роботов и робототехнических систем, включая введение разработанной модели в игровой процесс комплекса. Также реализуется процесс составления физических моделей робота, его электрических схем, а также процесс разработки, в том числе интеллектуального, программного обеспечения для выполнения задач управления роботом на тактическом и стратегическом уровнях.

Особенностью данного комплекса является разработка моделей робота, максимально приближенных к реальным экземплярам объектов. С помощью программного комплекса 3D Studio MAX происходит процесс создания геометрической модели, включающей установку внутренней структуры робота и обладающей массогабаритными характеристиками. После создания такой модели процесс переходит в сферу разработки электрической схемы робота, для осуществления подключения аппаратных компонентов и создания системы управления объектом [2]. Дальнейшим этапом разработки является использования механизма редактора структурных схем для создания программного обеспечения на борту разработанной модели.

## 1. 3. Gazebo

Gazebo представляет собой среду моделирования роботов, работающую в операционной системе Linux. Данный комплекс включает редактор моделей, позволяющий без дополнительного написания программного кода осуществить создание и редактирование сцены моделирования. Также особенностью данного редактора считается наличие встроенной библиотеки разработанных моделей, позволяющей в короткие сроки протестировать свое программное обеспечение. Популярными примерами роботов, находящихся в данной библиотеке являются iRobot Create, PR2, TurtleBot, Pioneer 2 DX. Помимо моделей роботов из библиотеки в комплексе также имеется возможность разработки дополнительных необходимых устройств.

Также стоит отметить, что комплекс предоставляет возможность для моделирования и симуляции работы сенсорной подсистемы роботов, к примеру камер, лазерных дальномеров, датчиков инерциальной навигации и др. Процесс описания модели реализован на специализированном языках SDF или URDF, которые, в свою очередь, являются надстройкой над XML.

Для осуществления симуляции физических процессов Gazebo оснащен следующими физическими движками: ODE, Bullet, Simbody, and DART [3].

Для удовлетворения потребностей к использованию хорошей графики, данный комплекс обладает возможностью к подключению объектно-ориентированного графического движка OGRE [3].

Очередной особенностью данного комплекса является его совместимость с кросс-платформенным программным обеспечением Player, что открывает возможности переноса отлаженного в Gazebo программного обеспечения на реальное оборудование. Также стоит отметить, что данная среда моделирования обладает совместимостью с ROS.

## 1. 4. Anykode marilou robotics studio

AnyKode Marilou Robotics Studio представляет собой среду для моделирования мобильных роботов, обладающую системой симуляции физических процессов. В качестве вводимых параметров для каждого объекта есть возможность задания его вращающих моментов, массы, свойств, и других. Для обеспечения симуляции физики используется движок Exec V5, который, в свою очередь, является многопоточным и кроссплатформенным.

AnyKode Marilou Robotics Studio поддерживается на операционных системах Linux Mint, Linux Ubuntu и OS Windows. Система Marilou обеспечивает возможность моделирования некоторого набора датчиков и приводов.

Также Marilou включает в себя SDK, обеспечивающие моделирование работы роботов и их компонентов MODA. После проведения синхронизации со средой программное обеспечение имеет возможность быть запущенным на другом устройстве, подключенном к сети. Доступные языки для программирования роботов насчитывают J#, VB#, C++ CLI, C/C++ и C# [4].

## 1. 5. V-REP

Среда моделирования V-REP является одной из современных кроссплатформенных сред и поддерживается в операционных систем Linux, Windows, Mac и отвечает всем требованиям современных платформ.

Процесс моделирования в V-REP можно организовать несколькими способами:

Первым способом является процесс выполнения программного обеспечения на отдельном вычислительном устройстве или на самом роботе, имеющим реализующим подключение к среде через реализацию шины. Достоинствами данного подхода является отделение контроллера от процесса моделирования и значительное уменьшение нагрузки на вычисление процесса моделирования. Однако он имеет и недостатки с синхронизацией цикла моделирования.

Второй способ заключается в выполнении симуляции на этом же устройстве, но в ином цикле моделирования. Недостатком является наличие задержек на переключение.

Третий способ предполагает выполнение кода программы и процесса моделирования в одном цикле. Достоинством данного способа является синхронизация процессов моделирования и программного обеспечения, что говорит об отсутствии каких-либо задержек. Недостатком является увеличение нагрузки на вычислительное устройство.

Для осуществления симуляции физических процессов V-REP оснащен следующими физическими движками: ODE, Bullet, Vortex и Newton [5].

Среда моделирование V-REP обладает двумя видами API:

1. Regular API позволяет вести разработку программного обеспечения в самой среде и предполагает написание скриптов на языках lua и C/C++.

2. Remote API обеспечивает взаимодействие со средой посредством сокетов. Данный API включает дистанционный сервер и Remote API клиентов, на которых обеспечивается реализация программного кода на языках Matlab & Urbi, Java, C/C++, Python. Запуск клиентов возможен на большом объеме оборудования, включающем роботов. Remote API реализовывает удаленное использование функций V-REP и обеспечивает передачу данных через поток.

Также стоит отметить, что данная среда моделирования обладает совместимостью с ROS, имеет в своем составе библиотеку с уже разработанными моделями роботов, элементов окружающей среды и моделями информационно-измерительной подсистемы с возможностью добавления и разработки своих вариантов моделей.

Среды визуального моделирования и симуляции широко используются в исследовательской и промышленной сферах разработки роботов и робототехнических устройств. На них лежат задачи тестирования и оптимизации программ, создания автоматизации производств и планирования. Также преимуществом использования данного подхода является избавление от излишних расходов на этапе разработки и отладки ПО. Также современные среды моделирования хорошо подходят для решения задач распознавания изображений, что является неотъемлемой частью задачи построения семантических карты.

# 2. Анализ подходов к представлению среды функционирования автономного мобильного робота

## 2. 1. OctoMap

Octomap это эффективная структура данных на основе октодеревьев, она представляет собой отображение сетки занятости, которое представляет окружающую среду ячейками сетки одинакового размера. Карта представляет собой пространство, разделенное на вокселы, где состояние каждого воксела сохраняется в узлах соответствующего октодерева. Это позволяет управлять разрешением карты путем ограничения глубины октодерева. Пример деления октодерева показан на рис. 1.

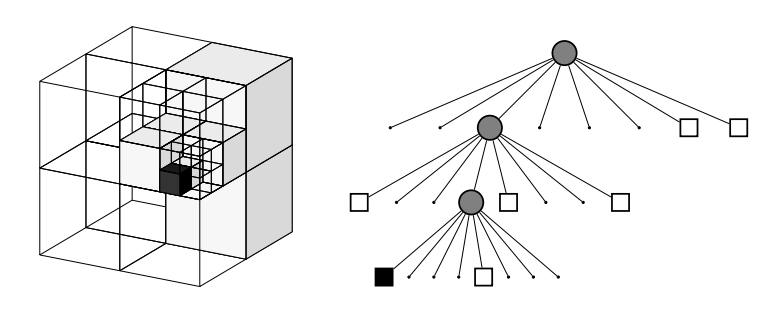


Рис. 1. Структура октодерева

Octomap фиксирует местоположения препятствий в окружающей среде, поддерживая вероятность занятости для каждой ячейки, которая обновляется независимо и постепенно с использованием байесовской фильтрации. Благодаря такому представлению данных OctoMaps сокращают использование памяти при отображении сетки занятости, организовывая ячейки в эффективную структуру данных на основе октодеревьев [5].

Также к преимуществам OctoMaps можно отнести то, что они могут создавать карты переменного разрешения из одних и тех же базовых данных.

Несмотря на преимущества, вышеупомянутые методы основаны на предположении, что все ячейки сетки являются статистически независимыми, а данные датчика связаны только с ячейками сетки, непосредственно пересекаемыми лучами дальности. В результате этого строгого допущения разреженные измерения датчиков приведут к прерывистым картам занятости между соседними показаниями датчиков или линиями сканирования, что может представлять угрозу для задач навигации, если планировщики пути считают зазоры незанятыми.

В данном подходе показания датчиков объединяются с использованием отображения сетки занятости. Вероятность листового узла n быть занятым при измерениях датчика оценивается в соответствии с

Пример построения Octomap изображен на рис. 2.

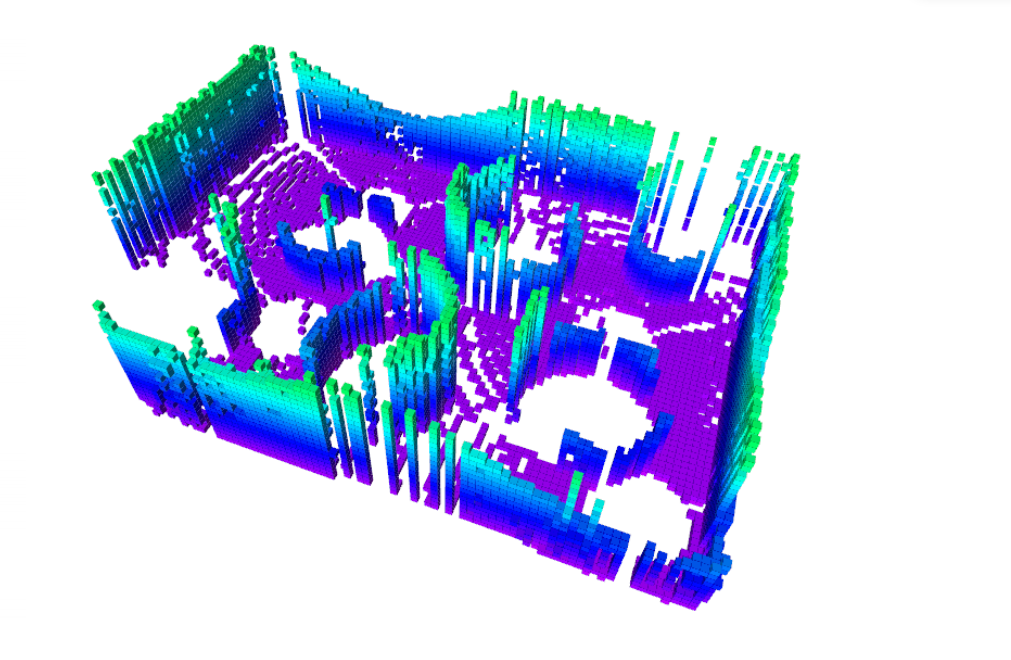


Рис. 2. Пример визуализации Octomap

## 2. 2. GPOctoMap

Gaussian Process Occupancy Mapping (GPOM) – алгоритм построения карт занятости с использованием гауссовских процессов. Данный подход картографирования используется для построения плотных карт занятости путем введения корреляции между соседними ячейками сетки занятости, основанной на использовании гауссовских процессов и непараметрического метода байесовского обучения, для вероятностного вывода в ненаблюдаемых областях карт. Таким образом, гауссовские процессы способны выражать богатую корреляцию между ячейками карты и данными датчиков [6].

Другими словами, в отличие от предположения, что все ячейки сетки независимы, и их единственная зависимость от пересекающих их лучей дальности, карта занятости гаусовских процессов собирает наблюдения датчиков и соответствующие метки (свободные или занятые) в качестве обучающих данных, а разряженные ячейки карты в качестве тестовых данных, которые связаны с обучающими данными использованием ковариационных функций. Таким образом, после выполнения регрессии мы получаем вероятность занятости каждой ячейки путем «сжатия» выходных данных регрессии в вероятности занятости, используя двоичные функции классификации.

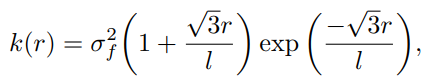
Для слияния множественных регрессий используется bayesian committee machines(BCM). Локальная карта занятости из нового сенсорного измерения неоднократно сливается с глобальной картой с использованием BCM.

Для оптимизации скорости вычислений предлагается использование вложенного поэтапного BCM, в котором выполняются два последующих слияния. После разделения новых наблюдений датчиков на несколько наборов обучающих данных и выполнения регрессии для каждого из них, на первом этапе объединения синтезируется результат нового наблюдения, в то время как второй этап объединяет полученный объединенный результат из нового наблюдения в существующую карту занятости. Это двухуровневое использование BCM уменьшает сложность регрессии, позволяя достичь трехмерного отображения в реальном времени с крупномасштабными наборами данных [7].

Вероятность занятости ячейки карты может быть выражена как:

где - это ячейка карты *i*; - набор наблюдений с датчиков, а - набор поз робота.

Ковариационные функции или функции ядра определяют сходство между парой точек: если они близки друг к другу, они, как правило, имеют одинаковое целевое значение. Чтобы уловить резкие колебания занятости, обычно возникающие в сценариях картографирования реального мира используется ковариационная функция Matern. Ковариационная функция Matern с параметром гладкости ν = 3/2 определяется как:



где , и два гиперпараметра и *l* являются предыдущей дисперсией сигнала и характеристической шкалой длины. Стоит отметить, что корреляция быстро падает с увеличением расстояния между точками. Основываясь на этом свойстве, для аппроксимации регрессии гауссовского процесса используется подмножество обучающих данных.

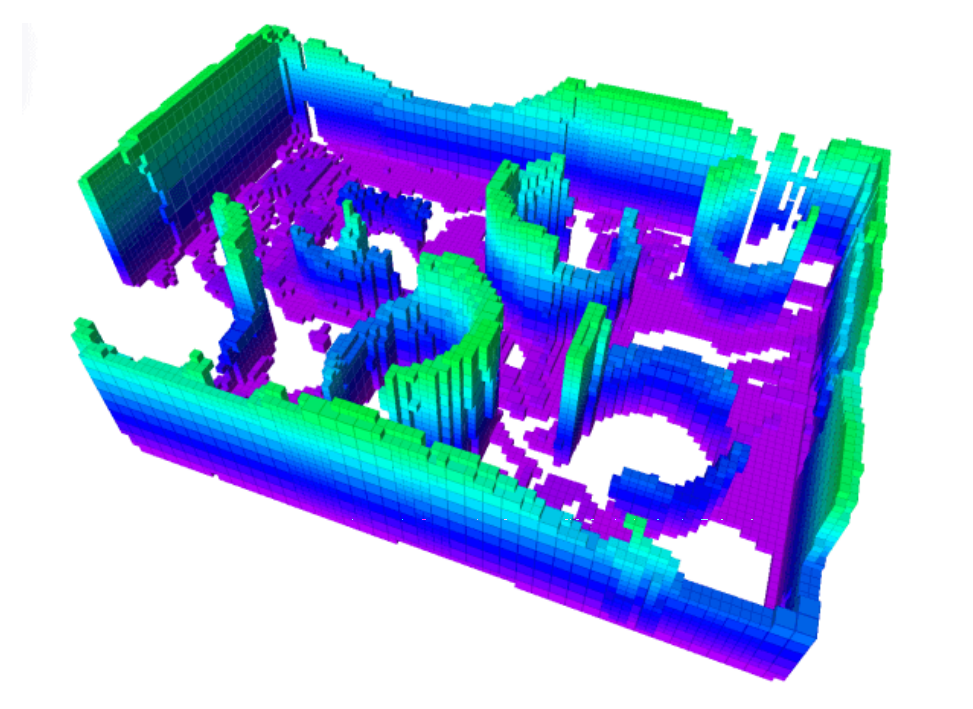


Рис. 3. Пример визуализации GPOctoMap

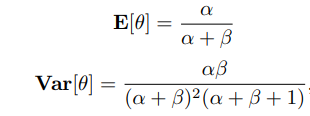
## 2. 3. BGKOctoMap

Поскольку bayesian committee machines (BCM) является приблизительным обновлением регрессии гауссовского процесса. Последовательное применение приблизительных обновлений может привести к ненадежному прогнозированию в сценариях долгосрочного картографирования и может привести к плохому выводу в ненаблюдаемых областях карты.

BGKOctoMap использует байесовский обобщенный вывод ядра для прогнозирования карты занятости и представляет собой модель, способную к точным рекурсивным обновлениям. Вместо того, чтобы пытаться делать все более неточные прогнозы, когда исследуемые точки находятся далеко от обучающих данных, метод обеспечивает точные прогнозы в областях, где имеется достаточное количество обучающих данных, и плавно переходит к некоторым предшествующим вероятностям занятости с высокой дисперсией, которые могут состоять из прогноза, представляющего неизвестное состояние.

Таким образом, BGKOctoMap воплощает собой подход к отображению карт занятости на основе логического вывода, в котором при отсутствии достаточных обучающих данных система возвращается к некоторым предшествующим знаниям, а не пытается делать прогнозы на основании ограниченной информации. Такое поведение системы позволяет избежать ситуаций, когда можно агрессивно прогнозировать содержимое незанятого пространства, а затем планировать высокоскоростную траекторию в областях, которые вместо этого следует рассматривать как неизвестные [8]. Пример построения карты BGKOctoMap представлен на рис. 4.

Как и в стандартной формулировке отображения сетки занятости, ячейка карты *m* занята с вероятностью и свободна с вероятностью где - координата, соответствующая центру ячейки сетки. То есть, занятость распределена по Бернулли с параметром θ = p (m = 1 | x ∗). Мы стремимся оценить параметр *θ* ячейки с центром в точке запроса . Для этого используется непараметрическая модель байесовского вывода для экспоненциальных семейств. При *θ ∼ Beta (α, β)* прогнозируемое среднее значение и дисперсия *θ* являются следующими:



где *α и β* - гиперпараметры. На каждом временном шаге мы получаем сканирование диапазона, содержащее трехмерные точки и соответствующие им метки, то есть вероятности занятости, составляющие обучающий набор.

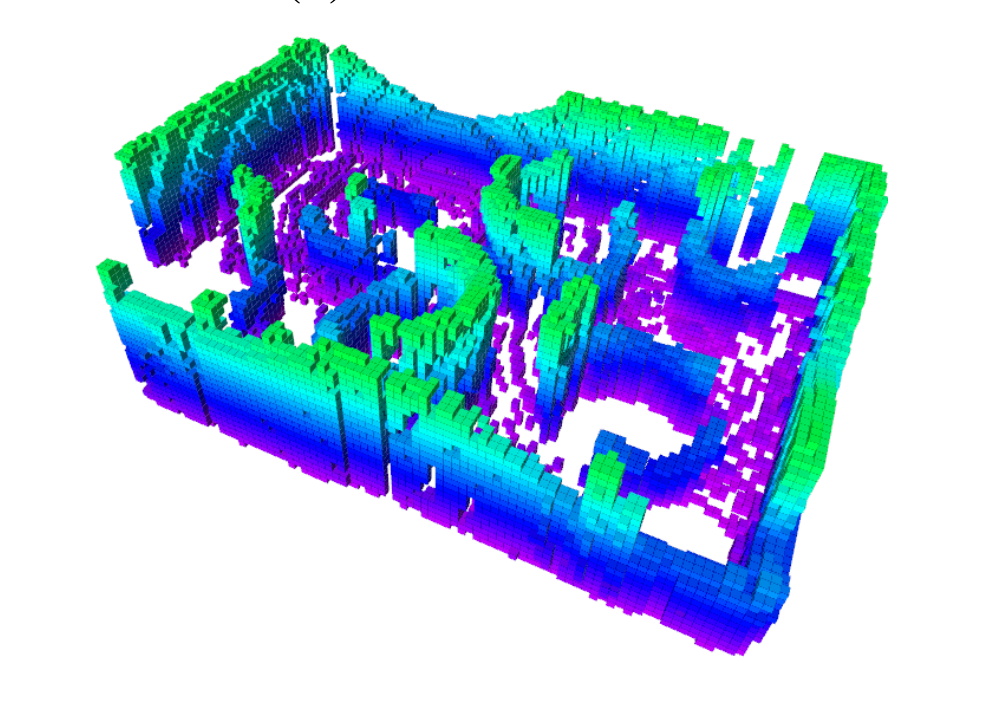


Рис. 4. Пример визуализации BGKOctoMap

Данный метод показывает отличные результаты, когда имеется несколько сканирований одних и тех же областей карты, поскольку он интегрирует новые данные более надежно, чем существующие методы отображения. Таким образов, данный подход эффективен в ситуациях динамического обновления одной карты за несколько сеансов либо группой роботов.

# 3. Разработка алгоритма построения семантической воксельной карты внешней среды автономного мобильного робота

Блок-схема алгоритма построения семантической карты представлена на рис. 5.

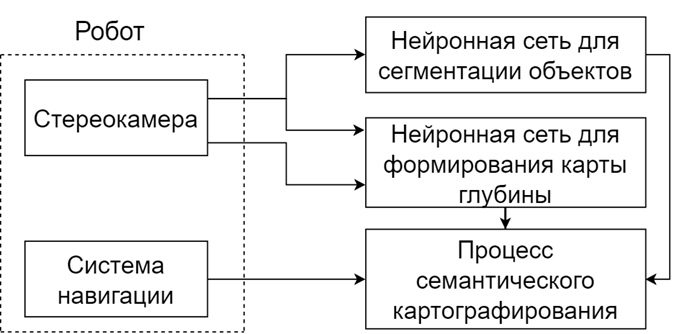


Рис. 5. Блок-схема алгоритма семантического картографирования

В разделе построения семантической карты используется расширение модели построения карты занятости BGKOctoMap для непрерывного семантического отображения, которая моделирует занятость и семантические вероятности классов окружающих объектов в единой структуре. Карта семантической занятости обладает свойствами, подобными карте сетки занятости, однако, поддерживает для каждой ячейки набор вероятностей семантических классов.

Данная модель описывает вероятность занятости с помощью функции правдоподобия Бернулли. Для каждой ячейки подсчитывается, как часто луч заканчивался в ней и как часто он проходил через нее. Эта модель имеет сопоставимые характеристики с байесовскими обновлениями для отображения сетки занятости. Модель семантического картографирования является ее естественным обобщением от отображения занятости (двоичного) до семантического (мультиклассового) отображения.

Пусть *K = {1, 2, ..., K}* - набор меток семантического класса. Для любой точки отображения  *∈ X* мы имеем набор измерений , где≥ 0.

Предполагая, что ячейки карты индексируются с помощью ∈ , ячейка *j-й* карты может принимать одну из *K* возможных категорий с вероятностью каждой категории, отдельно указанной как , где . Ячейка *j-й* карты с семантической вероятностью описывается категориальным распределением как:

Для устранения ограничения отображения сетки занятости, когда параметры для каждой ячейки связаны только с измерениями, которые непосредственно попадают в нее или проходят через нее, используется Bayesian Spatial Kernel Inference для преобразования модели дискретного семантического подсчета в непрерывную модель с учетом локальных корреляций на карте [9].

Семантическое 3D картографирование с использованием набора данных KITTI представлено на рис. 6.

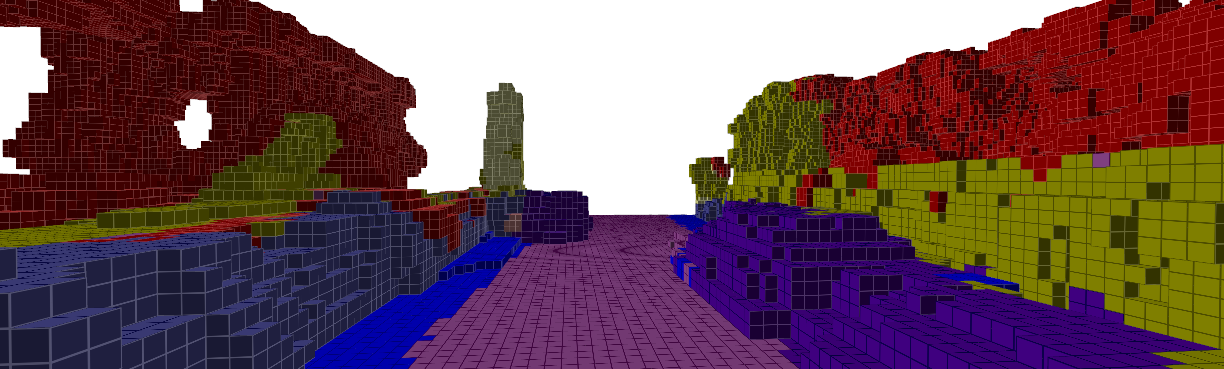


Рис. 6. Пример визуализации

# Заключение

Разработка данной системы построения семантических карт позволяет перейти к рассмотрению более сложного поведения системы, требующего понимания объектов окружающей среды. В свою очередь, использование октодеревьев для построения структурированного представления окружающего мира позволяет сократить использование памяти при отображении сетки занятости и организовать эффективную структуру данных для решения навигационных задач.

# Список использованных источников

1. Microsoft Robotics Developer Studio [электронный ресурс]

URL: http://microsoft.com/robotics

2. AnyKode [электронный ресурс]

URL: http://www.anykode.com/index.php

3. Gazebo [электронный ресурс]

URL: http://gazebosim.org

4. V-REP [электронный ресурс]

URL: http://www.coppeliarobotics.com

5. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees [электронный ресурс]

URL: http://www.arminhornung.de/Research/pub/hornung13auro.pdf

6. Fast, Accurate Gaussian Process Occupancy Maps via Test-Data Octrees and Nested Bayesian Fusion [электронный ресурс]

URL:http://personal.stevens.edu/~benglot/Wang\_Englot\_ICRA2016\_AcceptedVersion.pdf

7. Fast Gaussian Process Occupancy Maps [электронный ресурс]

URL: https://arxiv.org/pdf/1811.10156.pdf

8. Bayesian Generalized Kernel Inference for Occupancy Map Prediction [электронный ресурс]

URL:http://personal.stevens.edu/~benglot/Doherty\_Wang\_Englot\_ICRA\_2017.pdf

9. Bayesian Spatial Kernel Smoothing for Scalable Dense Semantic Mapping [электронный ресурс]

URL: https://arxiv.org/pdf/1909.04631.pdf

# Приложение А: Пример файла запуска воксельной карты ROS Gazebo Rviz

<launch>

<arg name="world\_file\_name" default="KITTI\_test" /> <!-- default="simple\_obstacle" />-->

<arg name="world\_path" default="$(find local\_planner)/../sim/worlds/$(arg world\_file\_name).world" />

<arg name="pointcloud\_topics" default="[/camera/depth/points]"/>

<node pkg="tf" type="static\_transform\_publisher" name="tf\_depth\_camera"

args="0 0 0 -1.57 0 -1.57 fcu camera\_link 10"/>

<include file="$(find local\_planner)/launch/local\_planner\_sitl.launch">

<arg name="model" value="kinect\_depth\_camera" />

<arg name="world\_path" value="$(arg world\_path)" />

</include>

<!-- Load custom console configuration -->

<env name="ROSCONSOLE\_CONFIG\_FILE" value="$(find local\_planner)/resource/custom\_rosconsole.conf"/>

<!-- Launch local planner -->

<node name="local\_planner\_node" pkg="local\_planner" type="local\_planner\_node" output="screen" >

<param name="goal\_x\_param" value="17" />

<param name="goal\_y\_param" value="15"/>

<param name="goal\_z\_param" value="3" />

<param name="world\_name" value="$(find local\_planner)/../sim/worlds/$(arg world\_file\_name).yaml" />

<rosparam param="pointcloud\_topics" subst\_value="True">$(arg pointcloud\_topics)</rosparam>

</node>

<node name="rviz" pkg="rviz" type="rviz" output="screen" args="-d $(find local\_planner)/resource/local\_planner.rviz" />

</launch>