

Геоинформатика. 2022. № 2. С. 25–33.
Geoinformatika. 2022;(2):25–33.

Применение ГИС-технологий

Научная статья
 УДК 004.9+681.5
<https://doi.org/10.47148/1609-364X-2022-2-25-33>

Повышение точности локализации беспилотной техники на основе электронной карты местности

© 2022 г. — Крюков Юрий Алексеевич^{1, а)}, Теряев Лев Николаевич^{1, б)}, Бобиков Станислав Алексеевич^{1, в)}

¹Государственный университет "Дубна"; Россия, г. Дубна
^{а)}kua@uni-dubna.ru, ^{б)}trchik228@gmail.com, ^{в)}stack@uni-dubna.ru

Аннотация: В статье рассматривается реализация алгоритма повышения точности локализации беспилотной техники на основе применения операционной системы для роботизированных устройств ROS. В рамках исследования поставлена задача определения точного положения объекта (роботизированного транспортного средства), которое без применения специальных средств отображалось неточно при использовании GPS. Поставленная в исследовании задача решалась при помощи ROS путем создания трехмерной карты местности и уточнению данных с GPS при помощи алгоритма.

Ключевые слова: операционная система робота, камера глубины, 3D карта, точность позиционирования

Для цитирования: Крюков Ю.А., Теряев Л.Н., Бобиков С.А. Повышение точности локализации беспилотной техники на основе электронной карты местности. // Геоинформатика. — 2022. — № 2. — С. 25–33. <https://doi.org/10.47148/1609-364X-2022-2-25-33>.

Application of GIS-technologies

Original article

Improving the accuracy of localization of unmanned vehicles based on an electronic terrain map

© 2022 — Yuri A. Kryukov^{1, а)}, Lev N. Teryaev^{1, б)}, Stanislav A. Bobikov^{1, в)}

¹Dubna State University; Dubna, Russia
^{а)}kua@uni-dubna.ru, ^{б)}trchik228@gmail.com, ^{в)}stack@uni-dubna.ru

Abstract: The article uses an algorithm for applying the probability algorithm for the occurrence of unmanned vehicles based on the use of a system for ROS robotic devices. As part of the study, the task was set to determine the exact position of an object (robotic vehicle), which, without special use of tools, was placed inaccurately in the GPS assembly. The task posed in the search was solved using ROS by creating a three-dimensional map of the area and refining the data using GPS using an algorithm.

Key words: robot operating system, depth camera, 3D map, positioning accuracy

For citation: Kryukov Yu.A., Teryaev L.N., Bobikov S.A. Improving the accuracy of localization of unmanned vehicles based on an electronic terrain map. *Geoinformatika*. 2022;(2):25–33. <https://doi.org/10.47148/1609-364X-2022-2-25-33>. In Russ.

Для решения многих задач необходимо учитывать реальное положение объекта в пространстве. При этом различные виды GPS-трекеров не всегда могут обеспечить достаточную точность (GPS не сможет определить положение объекта, который находится под навесом или крышей). В следствии чего необходимо, найти решение, которое помогало бы отслеживать положение робота в таких ситуациях.

Рассмотрим пример решения данной проблемы с помощью применения технологий ROS на роботизированной тележке.

Robot Operating System

ROS (*Robot Operating System*) — это операционная система для роботов, программа, которая предоставляет набор программных модулей для настройки множества различных сенсоров. В осно-

ве ее архитектуры лежит теория графов, которая позволяет создавать сложные и гибкие системы. Особенностью ROS также является обеспечение стандартных служб операционной системы, таких как: аппаратной абстракции, низкоуровневого контроля устройств, реализации часто используемых функций, передачи сообщений между процессами и управления пакетами.

Используемое оборудование

Система ROS сама по себе является очень мощным инструментом для разработки роботизированных продуктов. Однако практически в любой задаче необходимо получать информацию об окружающей среде, чтобы правильно обрабатывать различные ситуации.

Рис. 1. Строение камеры Intel RealSense d435i

Fig. 1. The structure of the Intel RealSense d435i camera

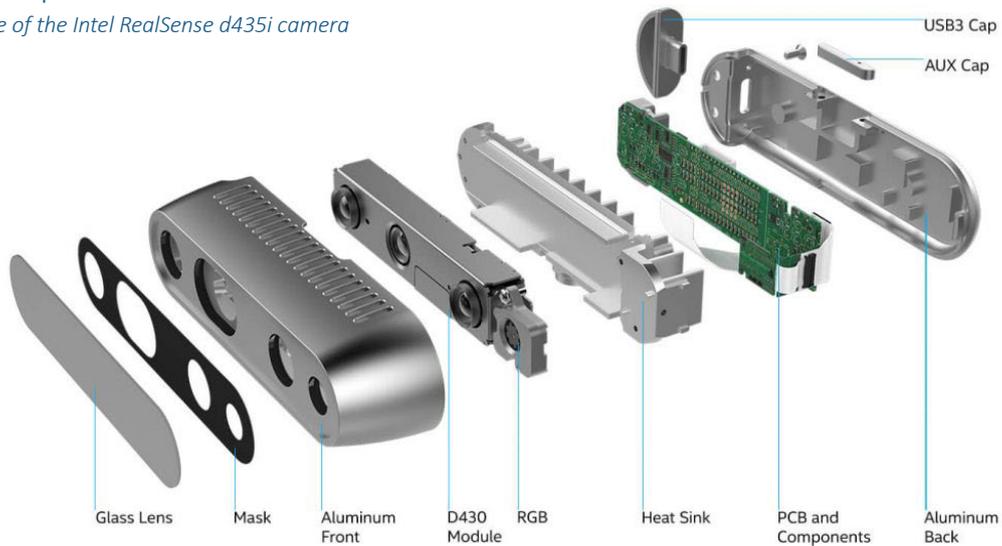
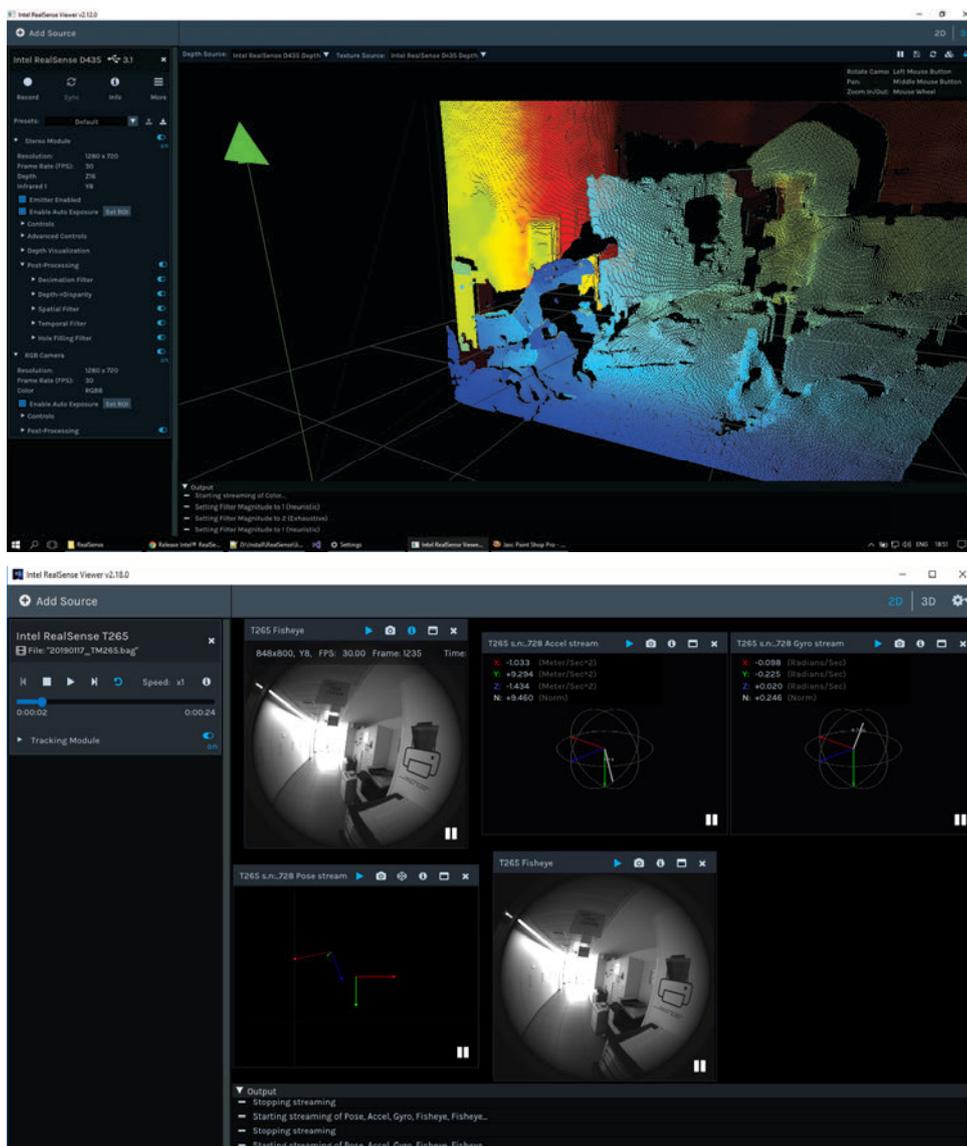


Рис. 2. Визуализация матрицы глубины

Fig. 2. Visualization of the depth matrix



Для решения задачи локализации в данном проекте были использованы следующие виды сенсоров:

- 1) Камера глубины *Intel RealSense D435i*.
- 2) Камера отслеживания *Intel RealSense T265*.

Intel RealSense D435i — устройство (рис. 1), которое позволяет оцифровывать информацию об окружающей среде. На выходе мы получаем облако точек в виде матрицы, каждый элемент которой — расстояние до некоторой точки реального объекта (рис. 2).

Стоит отметить еще ряд весомых качеств, благодаря которым было выбрано это оборудование. Определяющим преимуществом является возможность применения камеры не только в помещении, но и на улице. Модель *D435i* — имеет встроенный модуль *IMU (Inertial Measurement Unit)*, который позволяет отслеживать поворот и направление камеры в шести различных направлениях [4]. Тем более что *IMU* работает с фреймворком *ROS (Robot Operating System)* и, значит, может использоваться со всеми ROS-совместимыми платформами роботов. Данный сенсор является основой для процессов, которые будут описаны ниже [4].

Intel RealSense T265 — устройство, оснащенное двумя сверхширокоугольными объективами типа «рыбий глаз», шести-осевым инерционным датчиком *IMU* и специализированным *VPU (Visual Processing Unit) Intel Movidius Myriad 2* в качестве центрального вычислителя [3]. В отличие от других камер в линейке *Intel RealSense* данный сенсор на аппаратной базе реализуется набор алгоритмов, названных *Intel V-SLAM (Visual Inertial Odometry Simultaneous Localization and Mapping)*, одновременная локализация и картографирование на основе визуальной инерциальной одометрии). Данные алгоритмы позволяют значительно упростить задачу локализации робота [3].

Intel RealSense T265 — устройство, оснащенное двумя сверхширокоугольными объективами типа «рыбий глаз», шестиосевым инерционным датчиком *IMU* и специализированным *VPU (Visual Processing Unit) Intel Movidius Myriad 2* в качестве центрального вычислителя [1]. В отличие от других камер в линейке *Intel RealSense*, данный сенсор на аппаратной базе реализуется набором алгоритмов, названных *Intel V-SLAM (Visual Inertial Odometry Simultaneous Localization and Mapping)*, одновременная локализация и картографирование на основе визуальной инерциальной одометрии). Данные алгоритмы позволяют значительно упростить задачу локализации робота [1].

Теперь перейдем к разбору процесса локализации робота. В его основе лежат два важных под-

Рис. 3. Спецификация сообщения *Odometry*

Fig. 3. *Odometry message specification*

```
std_msgs/Header header
uint32 seq
time stamp
string frame_id
string child_frame_id
geometry_msgs/PoseWithCovariance pose
geometry_msgs/Pose pose
geometry_msgs/Point position
float64 x
float64 y
float64 z
geometry_msgs/Quaternion orientation
float64 x
float64 y
float64 z
float64 w
float64[36] covariance
geometry_msgs/TwistWithCovariance twist
geometry_msgs/Twist twist
geometry_msgs/Vector3 linear
float64 x
float64 y
float64 z
geometry_msgs/Vector3 angular
float64 x
float64 y
float64 z
float64[36] covariance
```

процесса: *Mapping* (построение карты местности на основании данных с датчиков) и *Localization*.

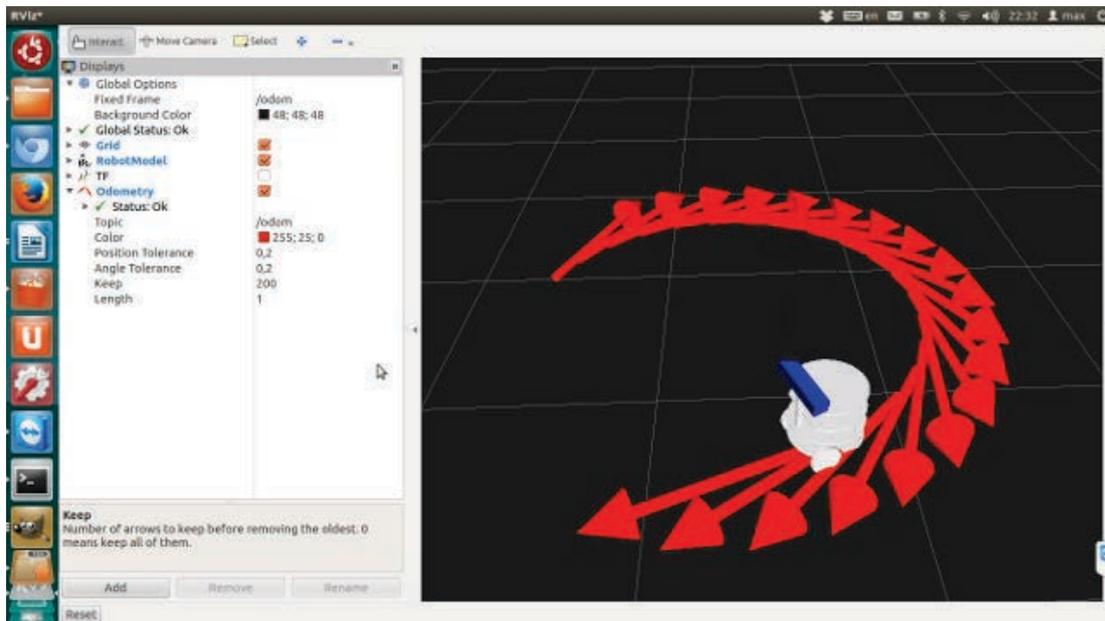
Mapping

Для того чтобы «робот» мог определить свое положение в пространстве, ему необходимо построить двумерную или трехмерную карту местности. Карта строится, исходя из набора следующих данных: *Odometry* и *LaserScan*.

Odometry — это специальный тип сообщений (рис. 3), который описывает положение «робота». Каждый раз при запуске процесса *Mapping*'а вокруг него формируется виртуальное пространство, в котором «робот» выступает точкой отсчета. А это значит, что *Odometry* будет содержать положение «робота» в метрах, относительно этой точки.

Сообщения о маршруте передвижения должен задавать пользователь, операционная система интерпретирует маршрут в команды управления двигателями. В случае с нашей роботизированной тележкой была использована камера *Intel RealSense T265*. За счет набора алгоритмов *Intel V-SLAM* (рис. 4) значения одометрии можно получать напрямую с камеры и перенаправлять их в нужный модуль.

LaserScan — еще один вид сообщений (рис. 5), который описывает расстояние до объектов [5]. Для

Рис. 4. Визуализация данных *Odometry*Fig. 4. Visualization of *Odometry* dataРис. 5. Спецификация сообщения *LaserScan*Fig. 5. *LaserScan* Message Specification

```
std_msgs/Header header
uint32 seq
time stamp
string frame_id
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
float32 range_min
float32 range_max
float32[] ranges
float32[] intensities
```

получения этих данных была использована камера глубины *Intel RealSense D435i*.

Однако на выходе получаются данные в формате *DepthCloud*, поэтому был написан специальный модуль, который конвертировал *DepthCloud* в *LaserScan* (рис. 6).

Когда появляется доступ ко всей необходимой информации, можно использовать модуль из стандартной библиотеки *ROS*, называющийся *GMapping*. Принцип работы модуля следующий: зная положение «робота» относительно начала координат и дистанцию до объектов в радиусе угла обзора камеры,

мы можем за несколько тактов алгоритма определить положение стен или препятствий в виде проекции на *2D* — плоскость.

На уровне системы создается новый процесс (узел) */slam_gmapping*, который подписывается на получение необходимых сообщений */scan* от камеры глубины и */tf* от камеры отслеживания (рис. 7, 8).

Localization

Localization — процесс определения положения робота. Данный процесс использует ранее созданную карту (сообщение типа *Map* [5]), сообщения типа *LaserScan* и *Odometry* [1].

С помощью модуля *Map* в систему загружается заранее подготовленная трехмерная карта местности (рис. 9). Для дальнейшего позиционирования робота на местности производится сопоставление данных, получаемых с датчиков, с информацией модуля *Map*. Модуль *map_server* после запуска начинает публиковать сообщения с данными о местности (см. рис. 10).

Используя *Map*, *Odometry* и *LaserScan*, можно приступить к процессу локализации. За него отвечает модуль *AMCL* (*Adaptive MonteCarlo Localization*) (рис. 10) [4]. Принцип его работы следующий: зная положение робота и расстояние до различных препятствий, мы можем сопоставить полученные значения с известными значениями на карте. Затем с некоторой вероятностью можно предположить несколько возможных позиций объекта в пространстве. Усредняя ранее полученные значения, можно в движении робота получать новые данные. Вследствие этого повышается точность позиционирования робота (рис. 11).

Рис. 6. Визуализация сообщений типа LaserScan
Fig. 6. Visualization of LaserScan messages

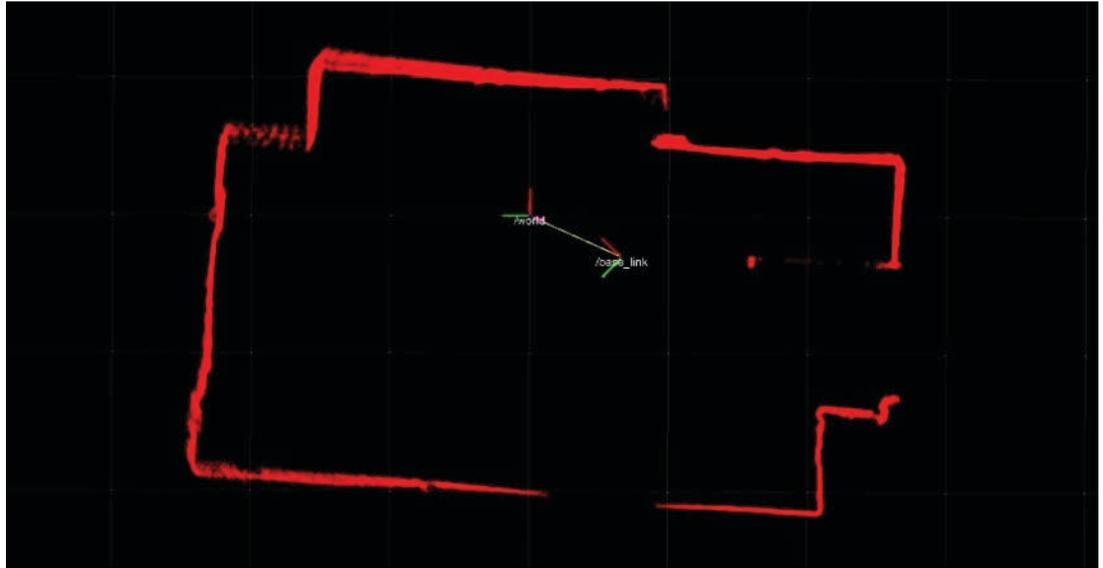


Рис. 7. Модуль GMapping
Fig. 7. GMapping module

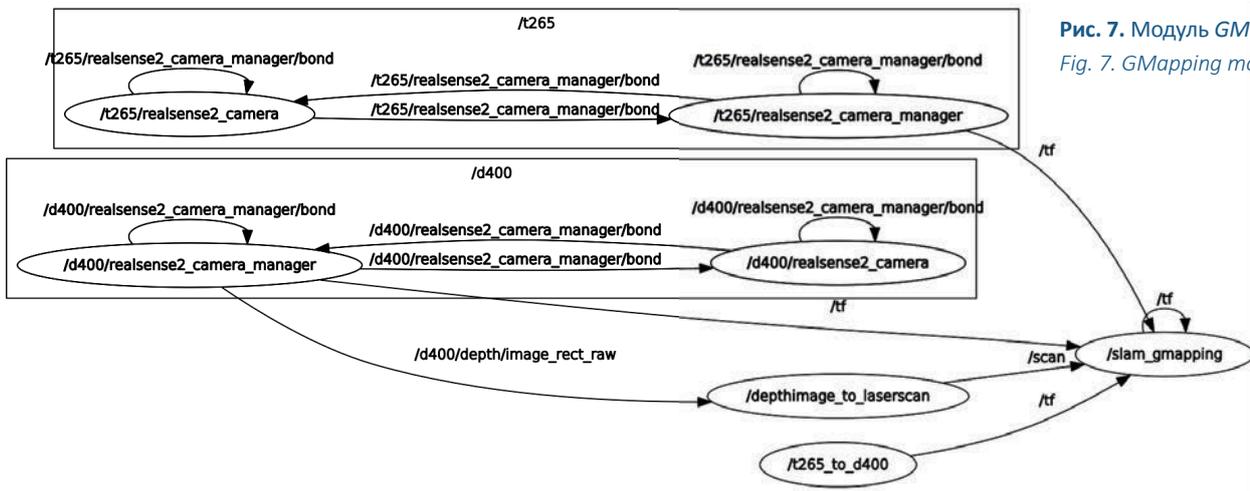


Рис. 8. Визуализация процесса построения 2D карты
Fig. 8. Visualization of the 2D map construction process

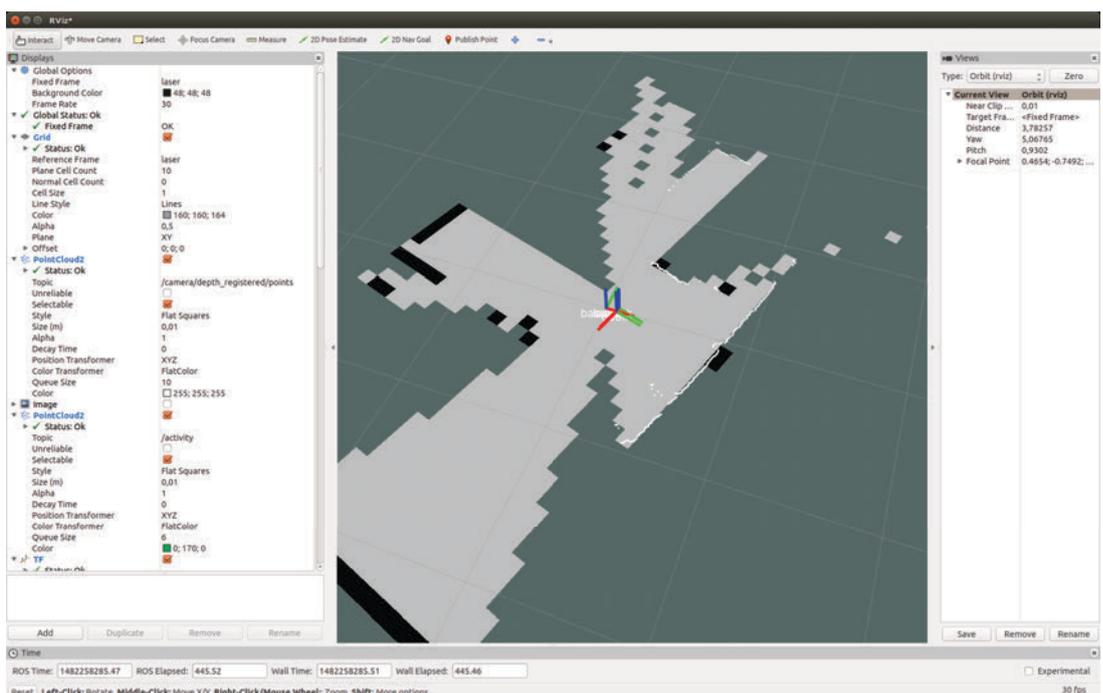


Рис. 9. Спецификация сообщения Map
Fig. 9. Map message specification

```
std_msgs/Header header
uint32 seq
time stamp
string frame_id
nav_msgs/MapMetaData info
time map_load_time
float32 resolution
uint32 width
uint32 height
geometry_msgs/Pose origin
geometry_msgs/Point position
float64 x
float64 y
float64 z
geometry_msgs/Quaternion orientation
float64 x
float64 y
float64 z
float64 w
int8[] data
```

Рис. 10. Визуализация процесса работы map_server
Fig. 10. Visualization of the map_server operation process

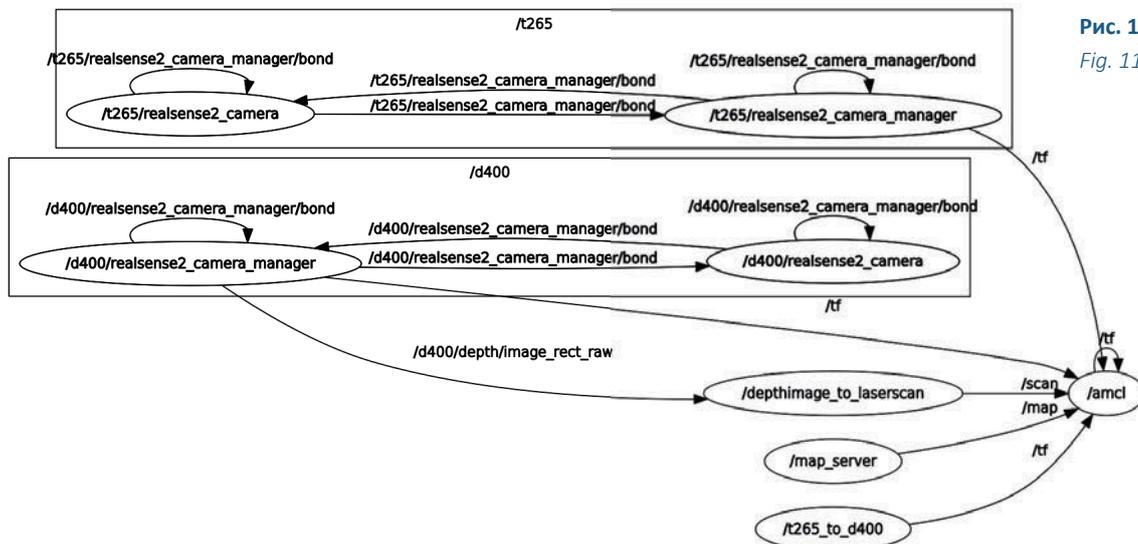
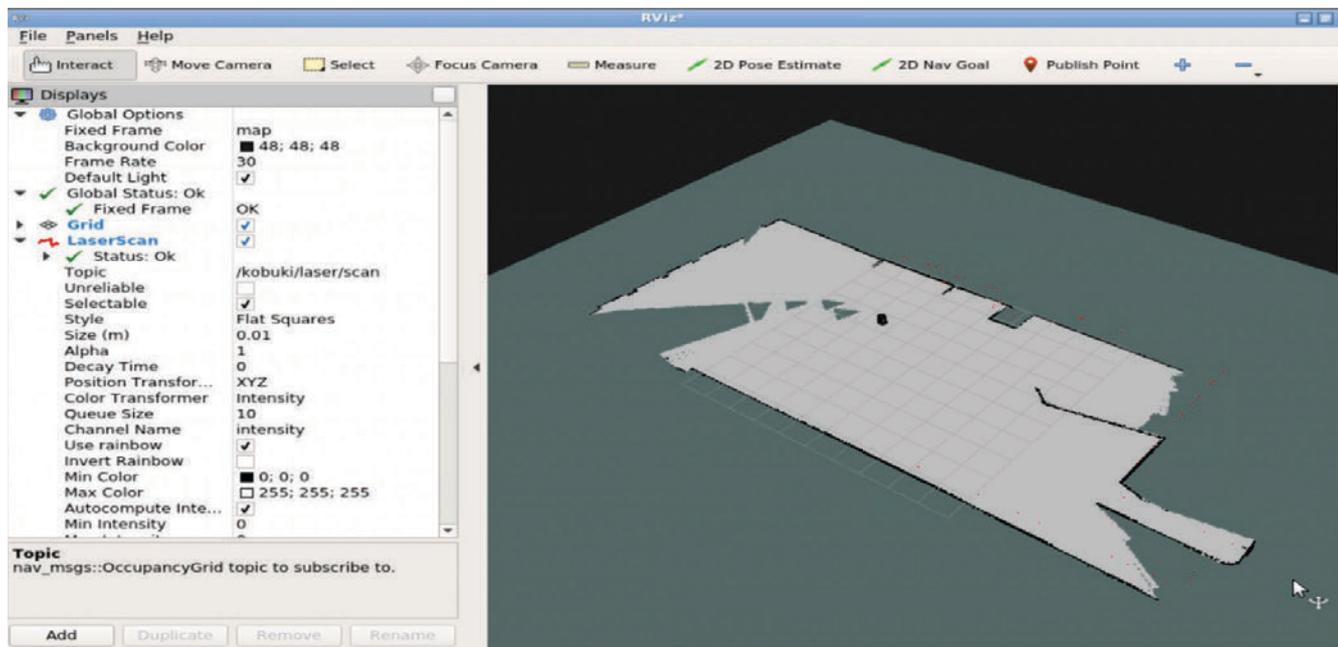
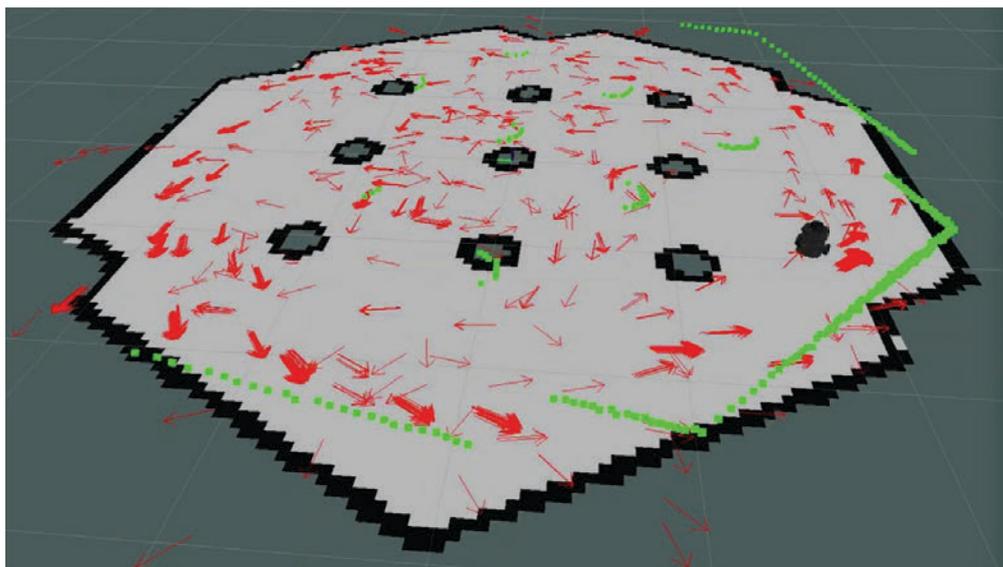


Рис. 11. Модуль AMCL
Fig. 11. AMCL module

Рис. 12. Визуализация процесса работы AMCL

Fig. 12. Visualization of the AMCL operation process



Уточнение GPS-позиции с использованием системы ROS

На данный момент все GPS-трекеры обладают одним недостатком. Когда транспортное средство находится под навесом, в туннеле и т.д., узнать точное положение объекта становится невозможно.

Как уже было описано ранее, система локализации, реализованная в ROS, позволяет создавать 3D карту местности [2]. С помощью данной карты можно определить положение робота в метрах от точки отсчета.

Таким образом появляется возможность создать некоторый алгоритм (рис. 13), который использует данные 3D карты, данные о положение робота

в локальном пространстве и данные, полученные с датчика GPS.

Постановка эксперимента и его результаты

На подготовленной площадке была нанесена разметка и добавлены места с устройствами подавления сигнала, имитирующие заезд робота на парковку или в туннель. При первом заезде использовался только модуль GPS. На участке трассы без устройств подавления сигнала точность позиционирования составляла 50–60 см и 90–110 см на поворотах. В зоне действия помех точность падает до 200–300 см и движение по GPS становится невозможным.

После загрузки геопривязанной карты на борт робота точность повысилась до 10–15 см при движении робота по прямой и 20–35 см на поворотах. При этом стоит отметить, что устройства подавления сигнала никак не влияли на точность позиционирования.

Особенностью данной технологии является, то что перед запуском программы, исполняемой роботом, необходима загрузка 3D карты для позиционирования на местности. Достаточно один раз загрузить большую карту местности, и далее робот сам будет дополнять её информацией с бортовых устройств при смене дорожной обстановки.

В конечном итоге были получены следующие результаты:

1. Точность позиционирования увеличилась в 3–3,5 раза.
2. Позиционирование может происходить в местах с плохой проходимостью сигнала.
3. Для корректной работы алгоритма необходимо задать начальное положение робота в GPS-координатах.

Рис. 13. Алгоритм уточнения позиции

Fig. 13. Position specification algorithm

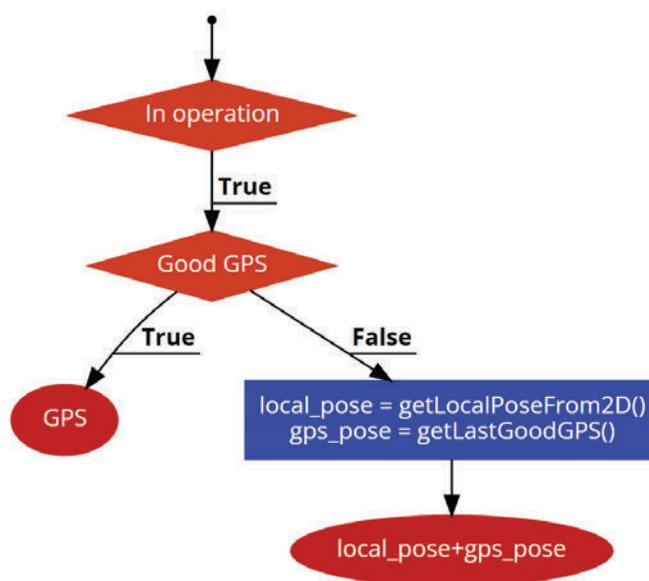


Рис. 14. Движение по GNSS в зоне действия помех

Fig. 14. GNSS movement in the interference zone



Рис. 15. Движение по 2D карте

Fig. 15. Movement by a 2D map



Заключение

Применение камер *Intel RealSense D435i* и *Intel RealSense T265* в совокупности с системами ROS может показывать хорошие результаты в процессе ло-

кализации «робота». Данная система может использоваться для решения большого спектра задач при организации беспилотного движения транспорта.

Список источников

1. Гурьев В. Intel RealSense Tracking Camera T265: совсем другой RealSense [Электронный ресурс] // Хабр. – 21.02.2019. – Режим доступа: <https://habr.com/ru/company/intel/blog/440826/> (дата обращения: 01.08.2021).
2. Титов А. ROS. Стек навигации [Электронный ресурс] // Хабр. – 03.05.2017. – Режим доступа: <https://habr.com/ru/post/327888/> (дата обращения: 01.08.2021).
3. 3D-камера Intel® RealSense™ D435i. Спецификации продукции [Электронный ресурс] / Intel. – 2018. – Режим доступа: <https://ark.intel.com/content/www/ru/ru/ark/products/190004/intel-realsense-depth-camera-d435i.html> (дата обращения: 01.08.2021).
4. ROS navigation stack. Package Summary [Электронный ресурс] // ROS wiki / Open Robotics. – 14.09.2020. – Режим доступа: <http://wiki.ros.org/navigation> (дата обращения: 01.08.2021).
5. ROS tutorials [Электронный ресурс] // ROS wiki / Open Robotics. – 03.06.2021. – Режим доступа: <http://wiki.ros.org/ROS/Tutorials> (дата обращения: 01.08.2021).

References:

1. Gurylev V. Intel RealSense Tracking Camera T265: sovsem drugoj RealSense [Intel RealSense Tracking Camera T265: a completely different RealSense]. In: Habr. 21.02.2019. Available at: <https://habr.com/ru/company/intel/blog/440826/> (accessed 01.08.2021).
2. Titov A. ROS. Stek navigacii [ROS. Navigation stack]. In: Habr. 03.05.2017. Available at: <https://habr.com/ru/post/327888/> (accessed 01.08.2021).

3. Intel® RealSense™ Depth Camera D435i. 2018. Available at: <https://ark.intel.com/content/www/us/en/ark/products/190004/intel-realsense-depth-camera-d435i.html> (accessed 01.08.2021).
4. ROS navigation stack. Package Summary. In: ROS wiki. 14.09.2020. Available at: <http://wiki.ros.org/navigation> (accessed: 01.08.2021).
5. ROS tutorials. In: ROS wiki. 03.06.2021. Available at: <http://wiki.ros.org/ROS/Tutorials>, free (accessed 01.08.2021).

Статья поступила в редакцию 22.02.2022, одобрена после рецензирования 01.03.2022, принята к публикации 02.03.2022.
The article was submitted 22.02.2022; approved after reviewing 01.03.2022; accepted for publication 02.03.2022.

Информация об авторах

Крюков Юрий Алексеевич

Кандидат технических наук, доцент
Проректор по научной и инновационной деятельности
Государственного университета «Дубна»
141982 Дубна, Московская обл., ул. Университетская, 19
e-mail: kua@uni-dubna.ru

Теряев Лев Николаевич

Аспирант,
директор Центра открытых образовательных технологий
Государственного университета «Дубна»
141982 Дубна, Московская обл., ул. Университетская, 19
e-mail: ttrchik228@gmail.com

Бобиков Станислав Алексеевич

Магистр,
специалист по учебно методической работе
Центра открытых образовательных технологий
Государственного университета «Дубна»
141982 Дубна, Московская обл., ул. Университетская, 19
e-mail: stack@uni-dubna.ru

Information about authors

Yuri A. Kryukov

Candidate of Technical Sciences, Associate Professor
Vice-Rector for Research and Innovation at Dubna State University
19 Universitetskaya str., Dubna, Moscow region, 141982, Russia
e-mail: kua@uni-dubna.ru

Lev N. Teryaev

PhD student,
Director of the Center for Open Educational Technologies at Dubna
State University
19 Universitetskaya str., Dubna, Moscow region, 141982, Russia
E-mail: ttrchik228@gmail.com

Stanislav A. Bobikov

Master's Degree,
Specialist in Educational and Methodological work of the Center for
Open Educational Technologies at Dubna State University
19 Universitetskaya str., Dubna, Moscow region, 141982, Russia
e-mail: stack@uni-dubna.ru