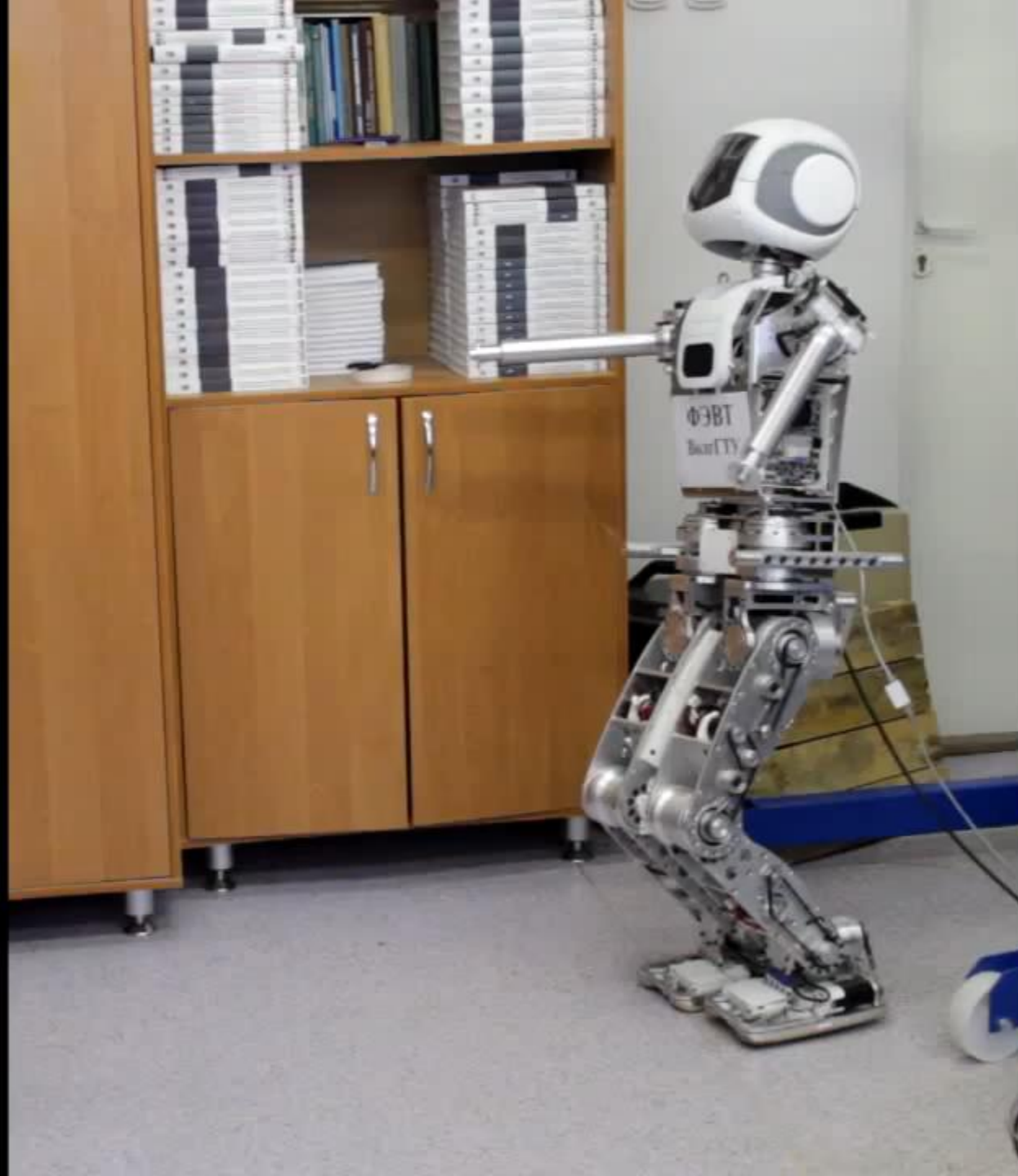


# Подсистема планирования траектории движения антропоморфного робота

Студент: Титов Алексей  
Группа: ИВТ – 460

Руководитель: Горобцов А.С.





## Цель

Разработать систему автономной навигации для антропоморфного робота AR600E и в частности, подсистему планирования траектории движения для робота

## Задачи

- 1) Анализ существующих решений, подходов и библиотек
- 2) Планирование архитектуры системы
- 3) Разработка:
  - Подсистемы планирования траектории движения робота
  - Подсистемы планирования параметров текущего шага робота (Марков А.Е.)
  - Коммуникации между системой компьютерного зрения и системой управления роботом

# Задача автономной навигации

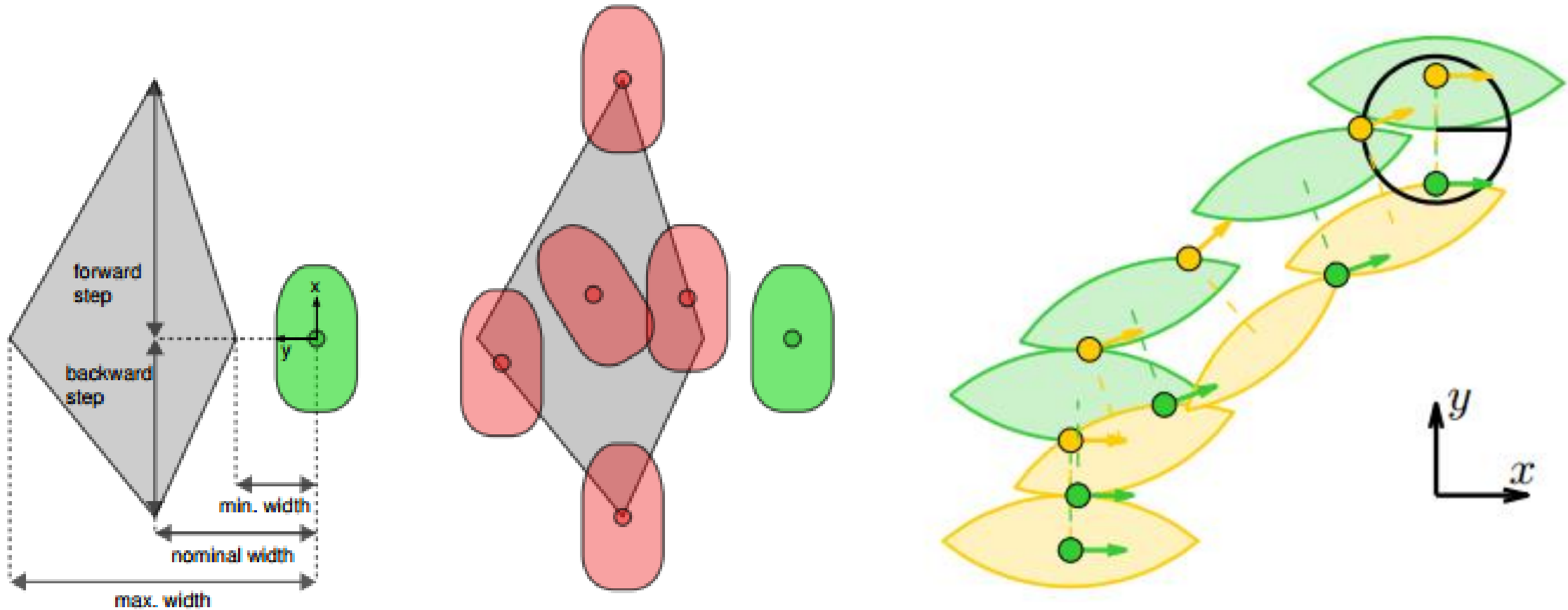
Задача получения информации об окружении  
(облако точек окружения и местоположение)

Задача планирования  
движений с учетом окружения

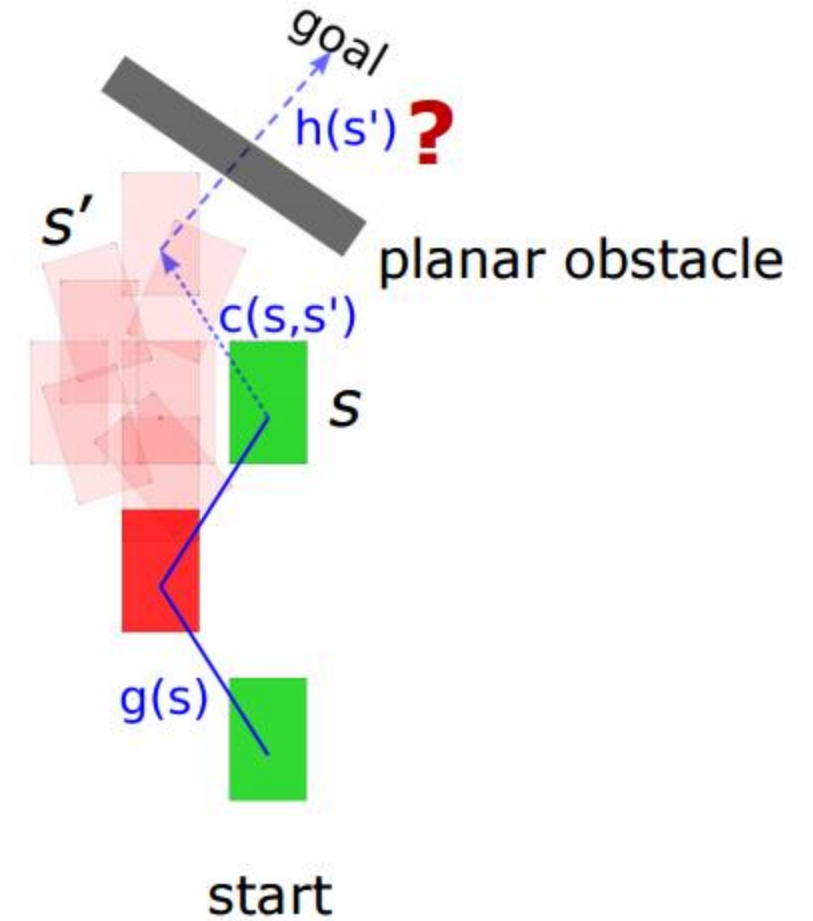
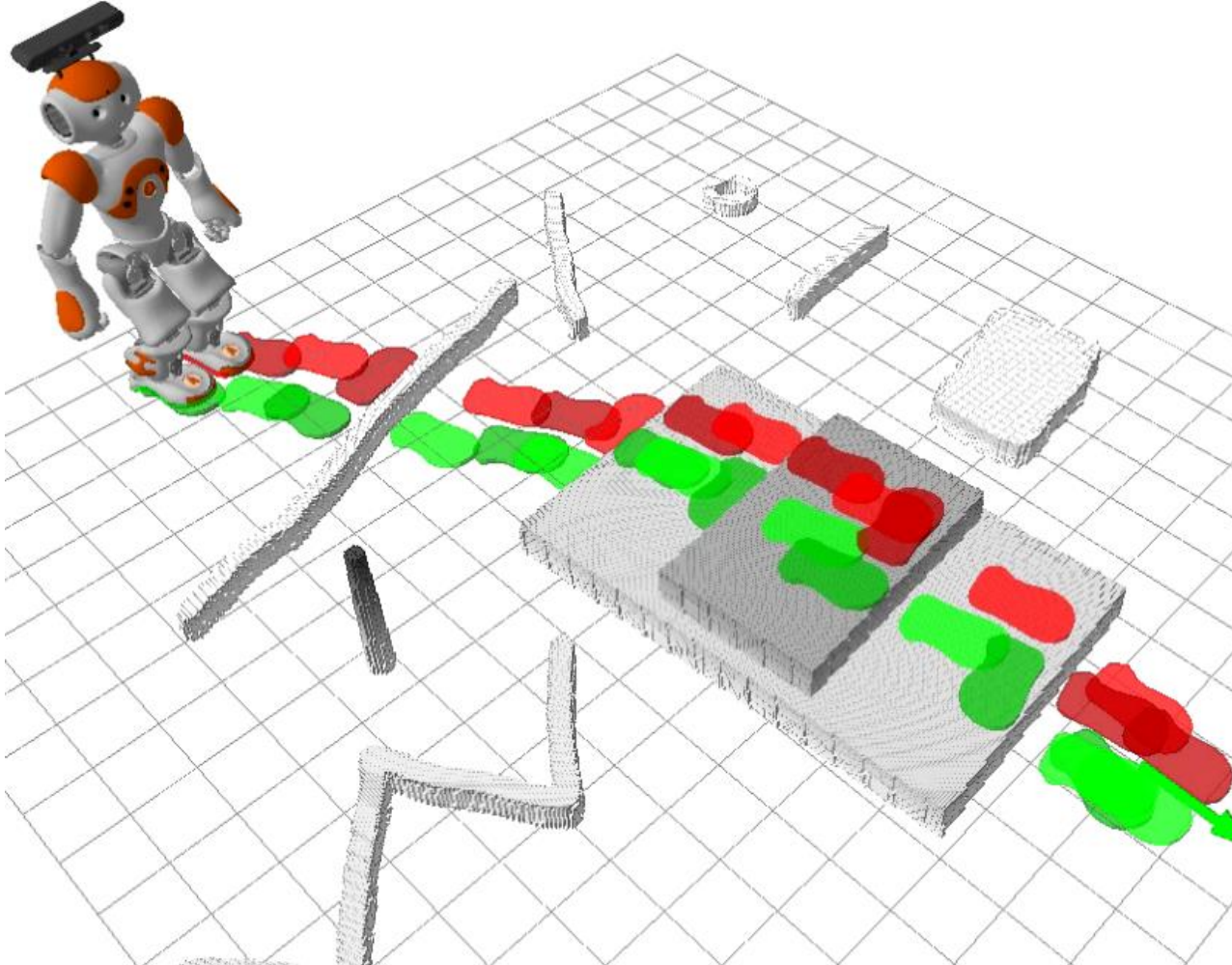
Задача планирования  
маршрута движения робота

# Обзор аналогов

# MIT DARPA Robotics Challenge Team

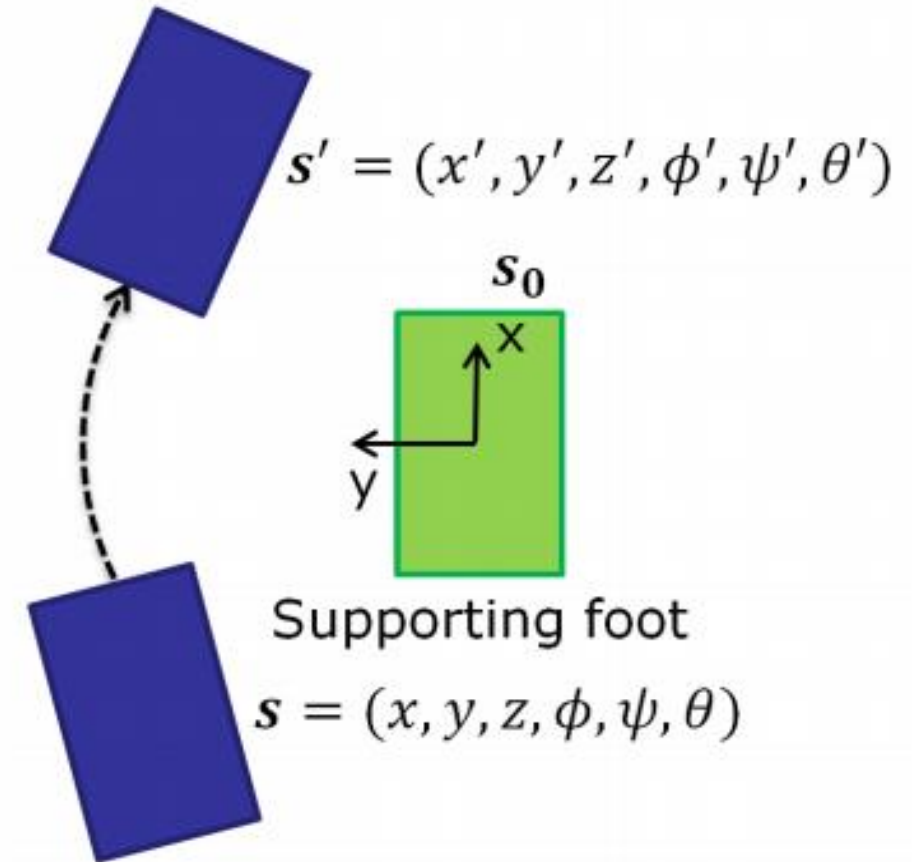
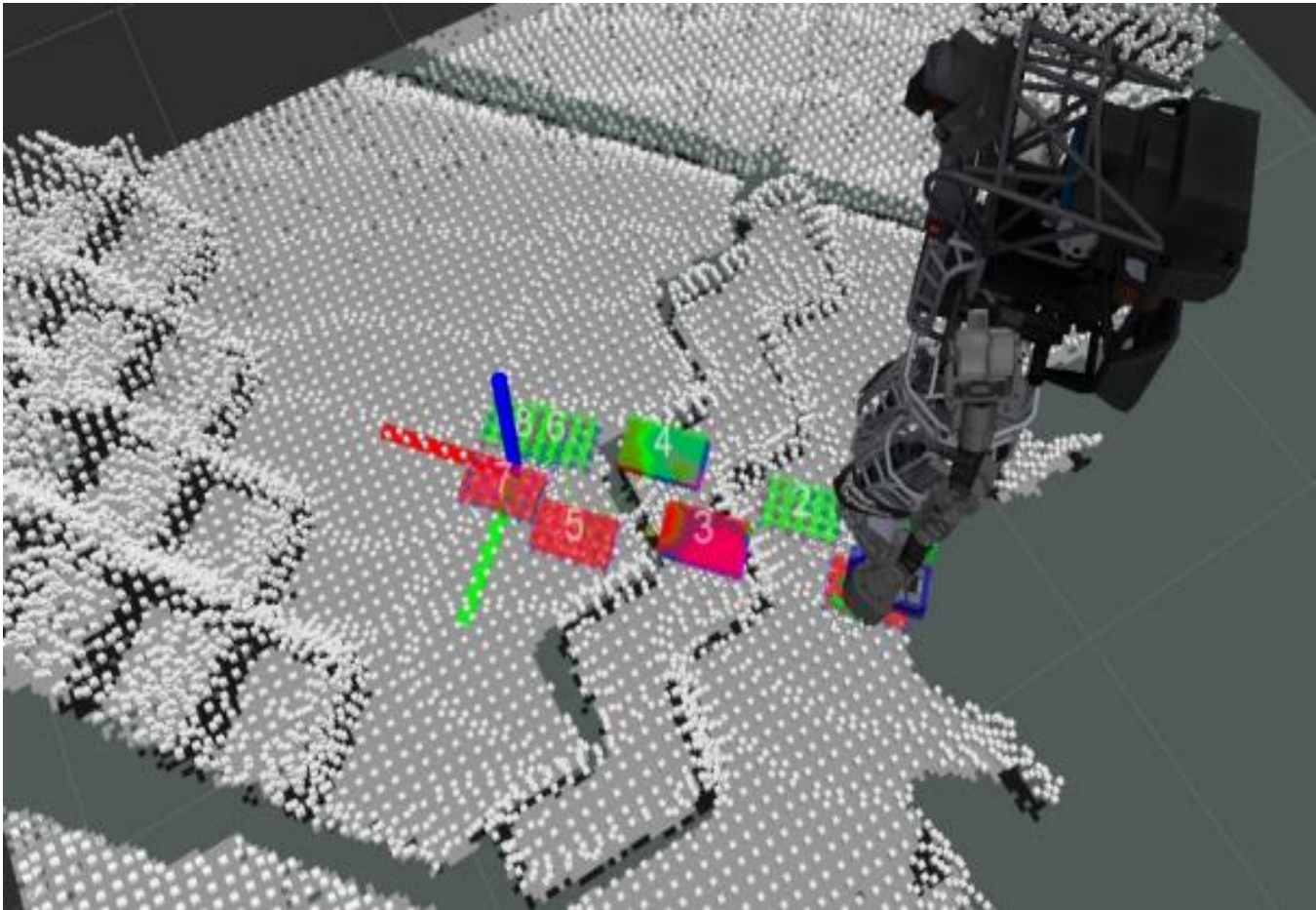


# Humanoid Robots Lab





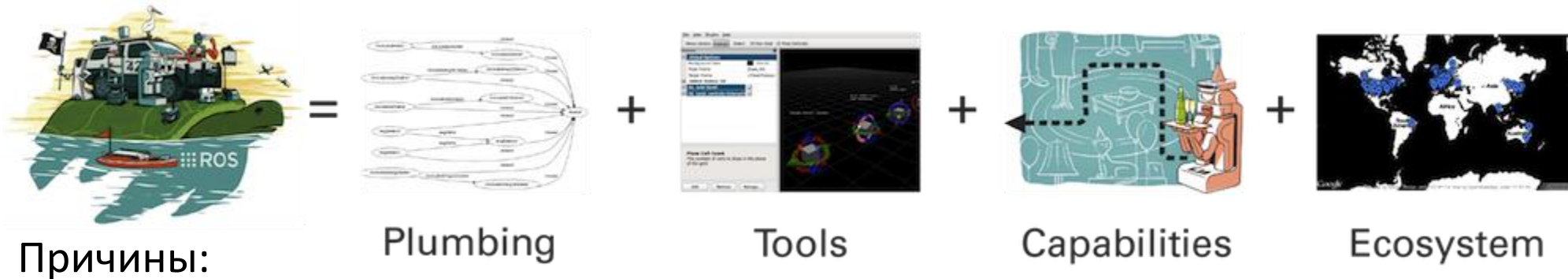
# Vigir footstep planner



# Разработка архитектуры системы автономной навигации

# О ROS

В качестве платформы для разработки был выбран фреймворк ROS на ОС Ubuntu.



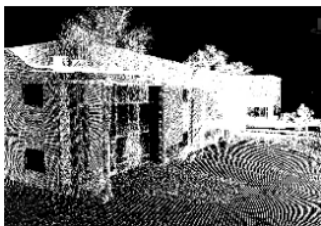
- Предоставляет много нужных пакетов для робототехники
- В частности хорошо развиты стеки:
  - Навигации
  - Локализации
  - Картографии (SLAM алгоритмы)
  - Планирования движений и маршрутов
- Предоставляет визуализаторы, удобную концепцию сообщений и подписчиков и много другого.
- <http://www.ros.org/core-components/>

# Варианты архитектур системы автономной навигации

Вариант 1 - Шаги генерирует ФРУНД

## SLAM алгоритм

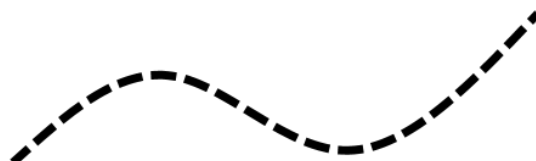
Предоставляет карту окружения и аппроксимацию местоположения в ней



$(x, y, z, \psi, p, r)$

## Глобальное планирование траектории

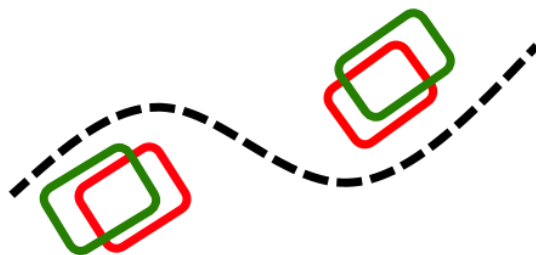
Предоставляет ФРУНДУ траекторию в виде линии (набора точек)



Планирование на 2D карте препятствий может упускать некоторые особенности поверхности

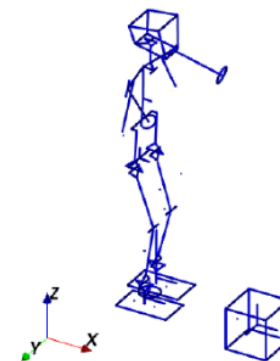
## Корректировка параметров шага

Корректирует шаг, если на месте требуемого положения ступни обнаружены препятствия. Сообщает, если шаг невозможен



## ФРУНД

- 1) Запрашивает траекторию в виде набора точек
- 2) Генерирует шаги так, чтобы не сильно отклоняться от траектории
- 3) В процессе планирования шагов ФРУНДОМ каждый шаг "проверяется" путем запросов к модулю планирования параметров шага. Эти запросы позволяют скорректировать шаг, если в требуемой точке обнаружены препятствия. Или остановить робота, если шаг невозможен



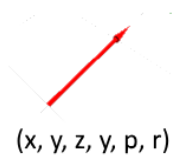
Frame: 0.0  
Time: 0.0

# Варианты архитектур системы автономной навигации

Вариант 2 - Шаги планирует система автономной навигации

## SLAM алгоритм

Предоставляет карту окружения и аппроксимацию местоположения в ней

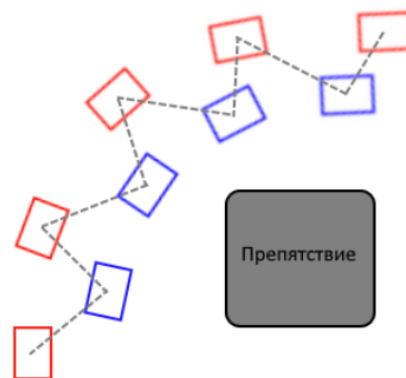


Карта окружения

Местоположение

## Модуль планирования маршрута

Генерирует маршрут в виде набора положений ступней, которые должен выполнить робот, чтобы достичь цели

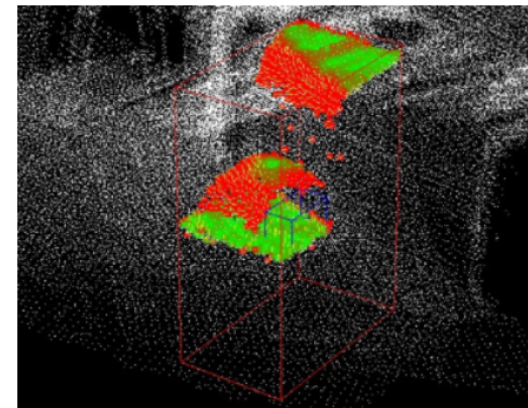


Запрос на проверку шага или на создание карты высот

Ответ

## Модуль корректировки параметров шага

Запрос о проверке планируемых шагов или построение карты весов / высот



Задание цели планирования

Спланированный маршрут

## ФРУНД

Запрашивает траекторию в виде набора положений ступней. Проверая устойчивость положений, генерирует движения так, чтобы повторять полученные шаги





# Рассмотрение модулей системы автономной навигации

Задача получения информации об окружении





# SLAM (с англ. одновременная локализация и картография)

Для автономной навигации, да и вообще чего - либо, требующего полной информации об окружении нужна карта самого окружения.

Для этих задач хорошо подходит карта в виде облака точек.

Vision-based SLAM алгоритмы позволяют строить карту окружения и приблизительно оценивать местоположение в ней.

На входе:

- Данные с датчиков, помогающих оценить местоположение (ИНС, любые данные одометрии)
- Данные с Vision датчиков (дальномеры, лидары, стереокамеры)

На выходе:

- Карта окружения
- Аппроксимация местоположения

В качестве **SLAM** алгоритма была выбрана реализация библиотеки **rtabmap**.

- Качественно задокументирована
- Поддерживает многие датчики в качестве источников данных (см. далее)
- В меру требовательна к ресурсам даже при создании больших карт (хватает ноутбука с Intel Core i3 + 4 Гб RAM)
- Имеет множество настроек и легко расширяема
- Предоставляет много доп. функций (например карту препятствий)
- Интеграция с библиотекой Octomap (спец. структуры для хранения и обработки плотных облаков точек)

Имеется также ряд других реализаций:

- **hector\_slam** (строит только 2D карту препятствий)
- **RGBDSLAMv2** (глубокая beta и слабая документация)
- **Kinect Fusion, PCL KinFU, ElasticFusion** (строят меш, высокие требования к ресурсам, малый размер карты)

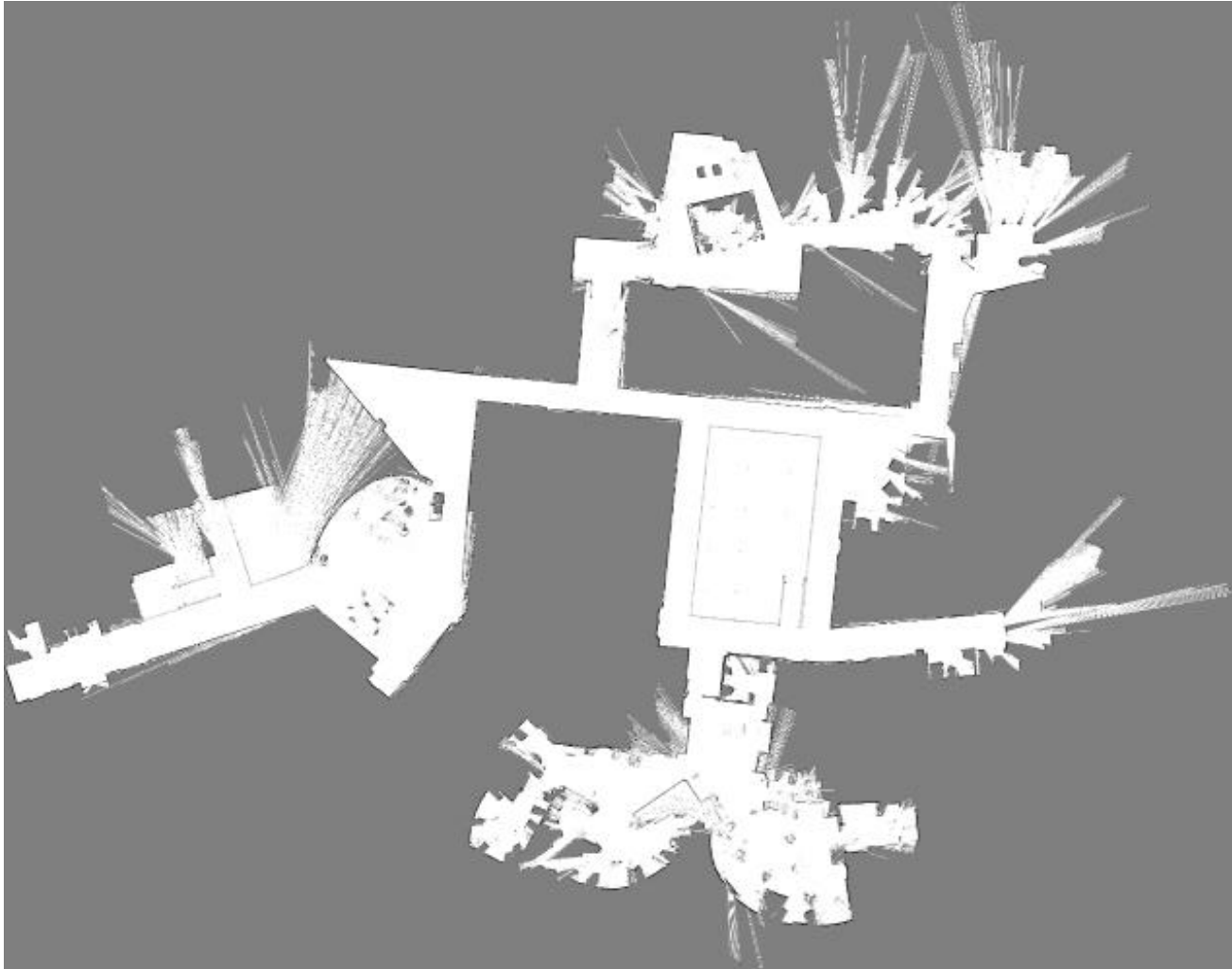


# Подходы к решению задачи планирования маршрута

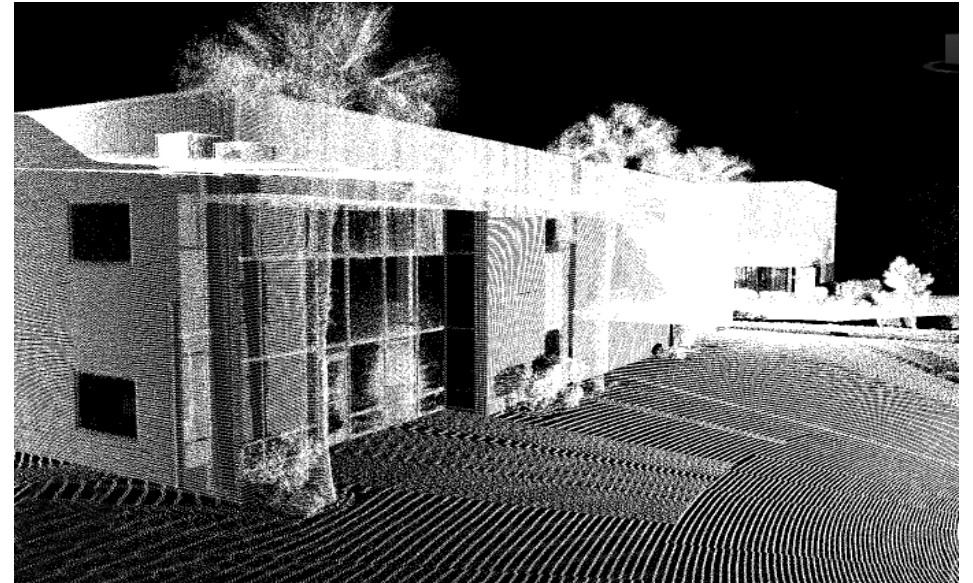


# Базовые структуры «карт»

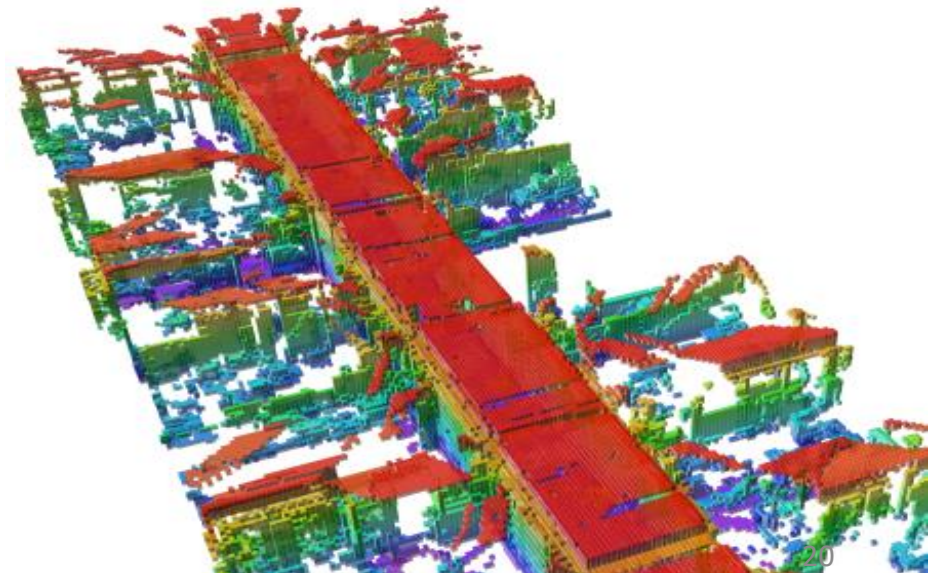
## Карта препятствий (Occurance Grid)



## Облако точек



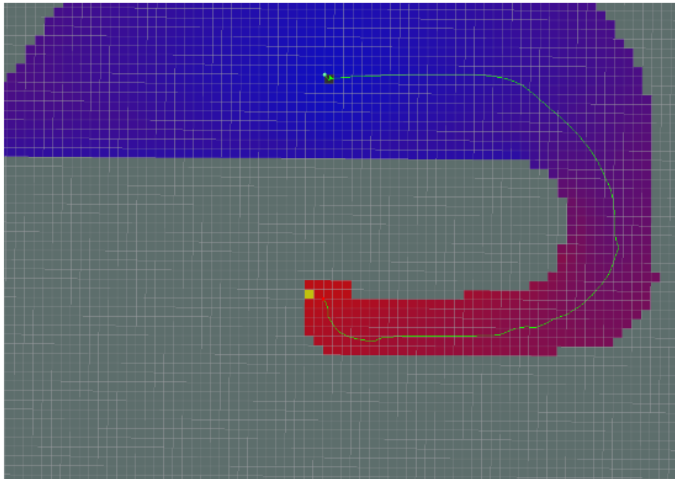
## K-D дерево



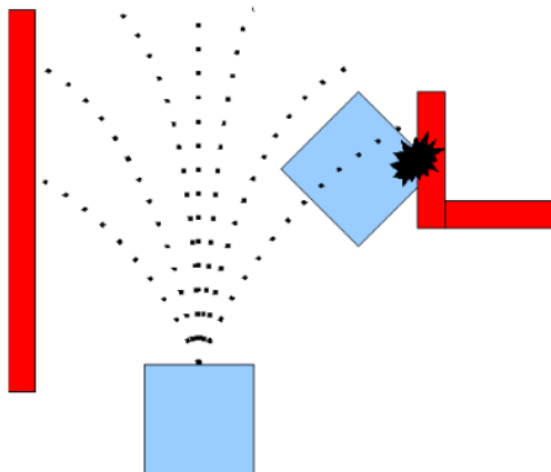
# Планирование на основе карты препятствий

## Move\_base

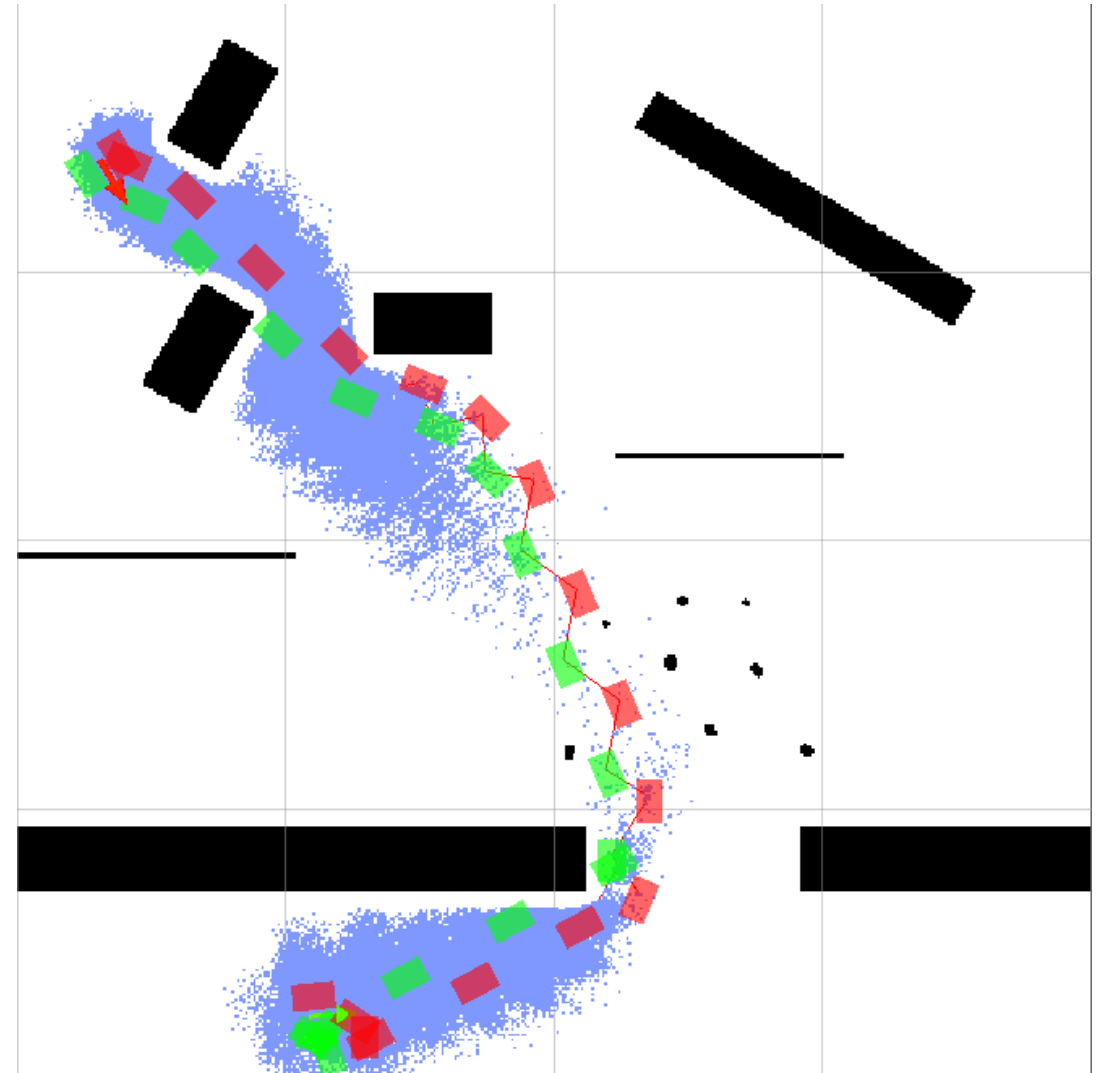
Global planner



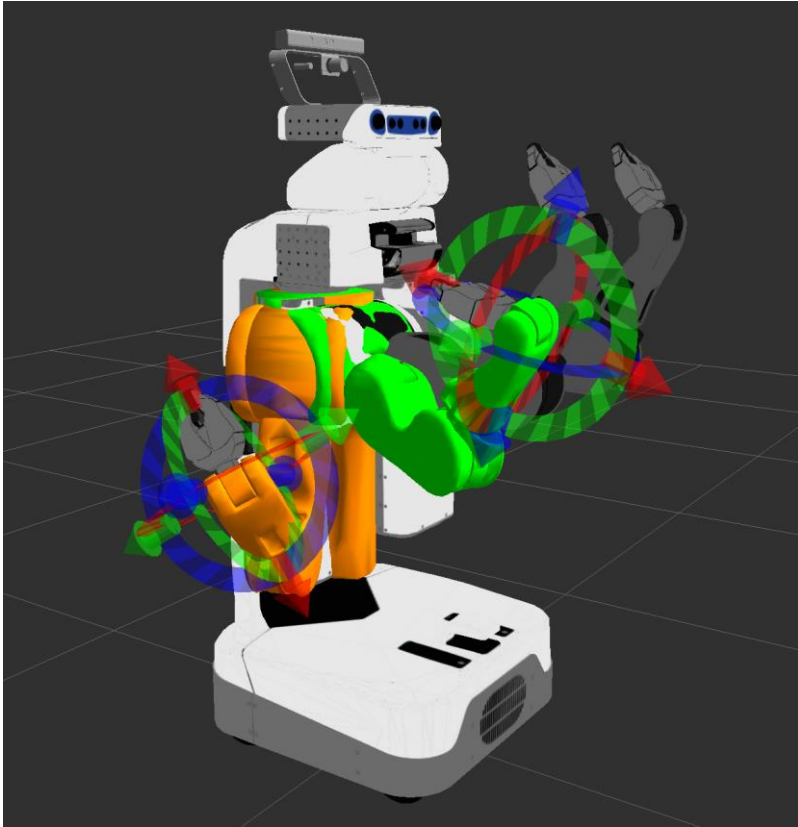
Local Planner



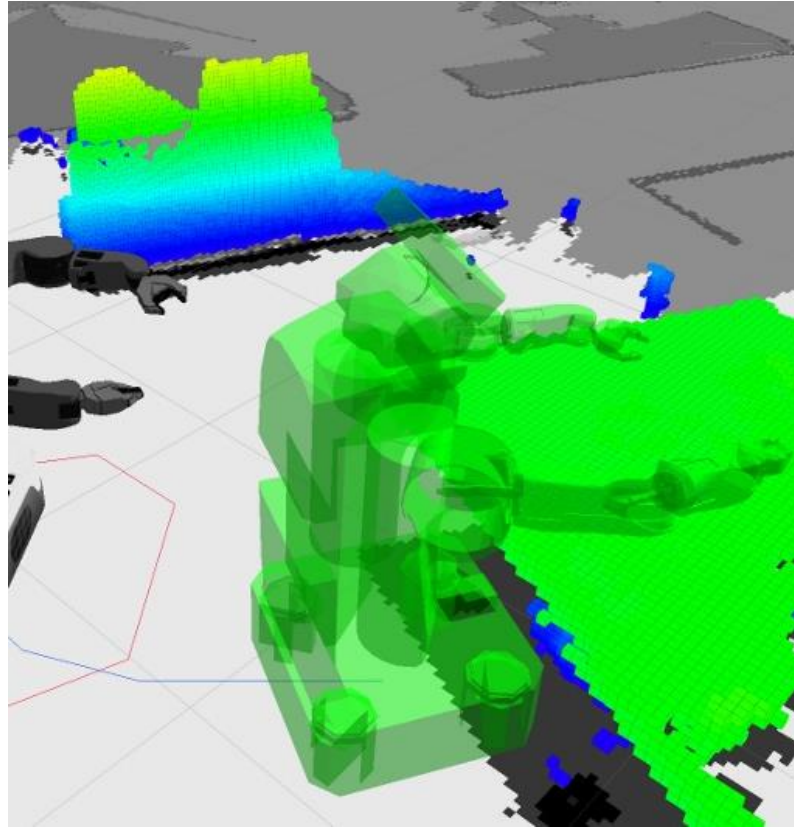
## Footstep\_planner



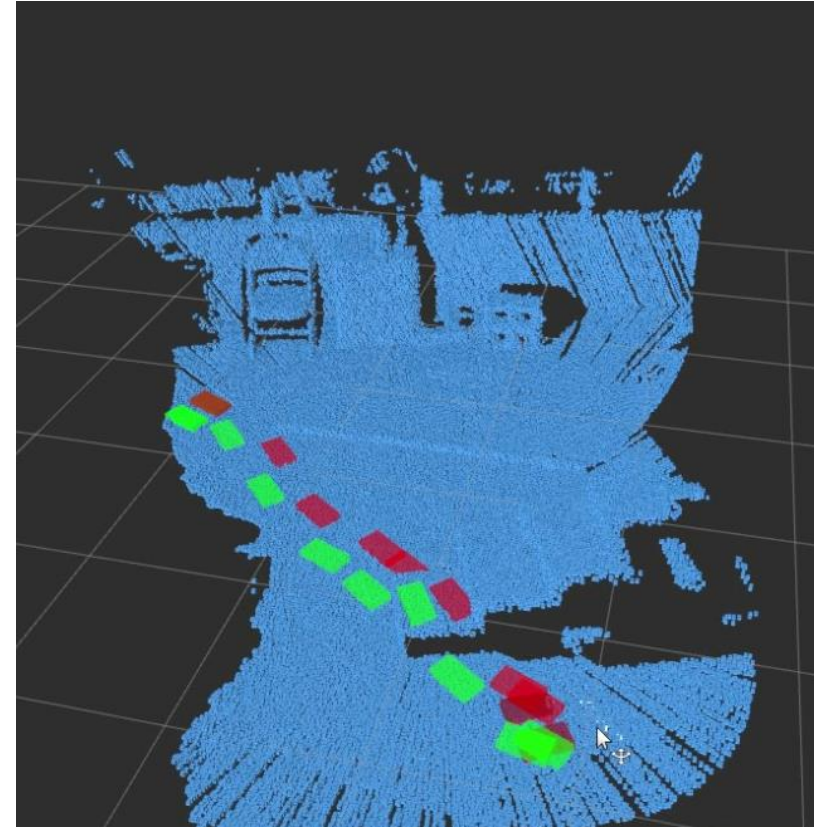
# Планирование на основе плотного облака точек



**Movelt**



**3d navigation**



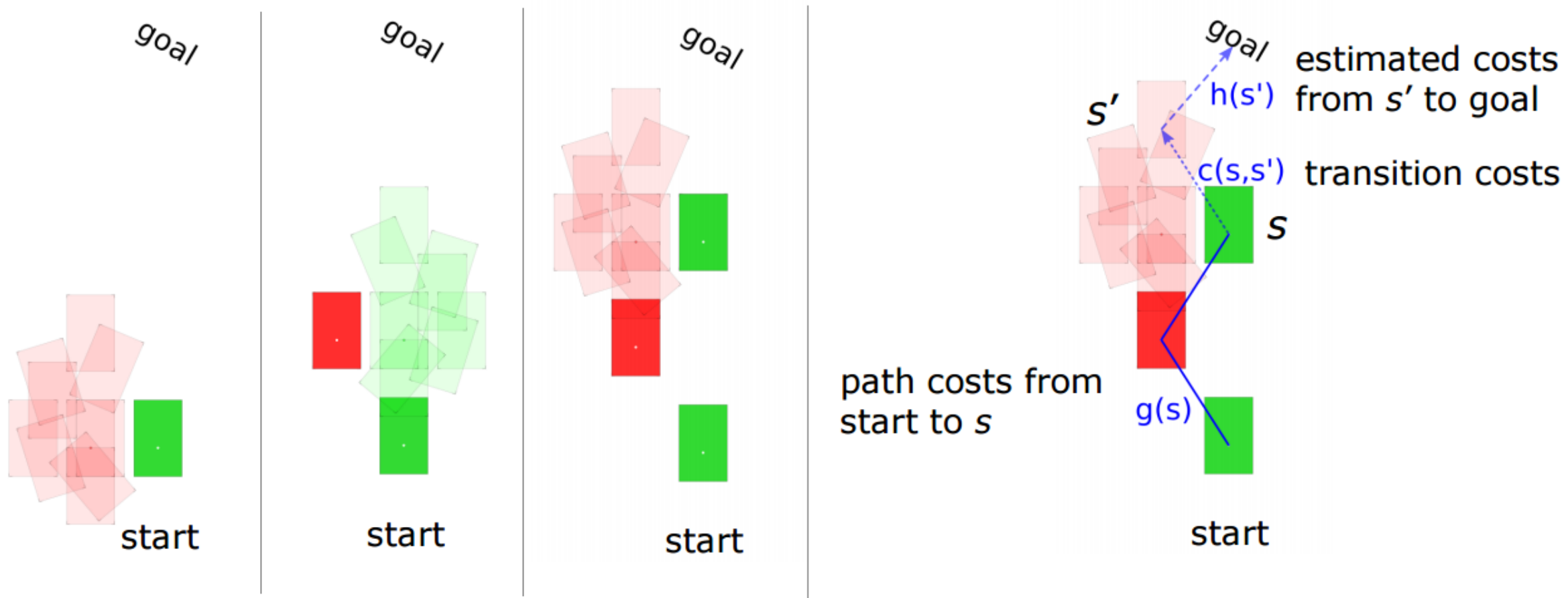
**Vigir footstep planner**

# Реализация модуля планирования маршрута

Вариант 1. Планирование маршрута в виде набора положений ступней

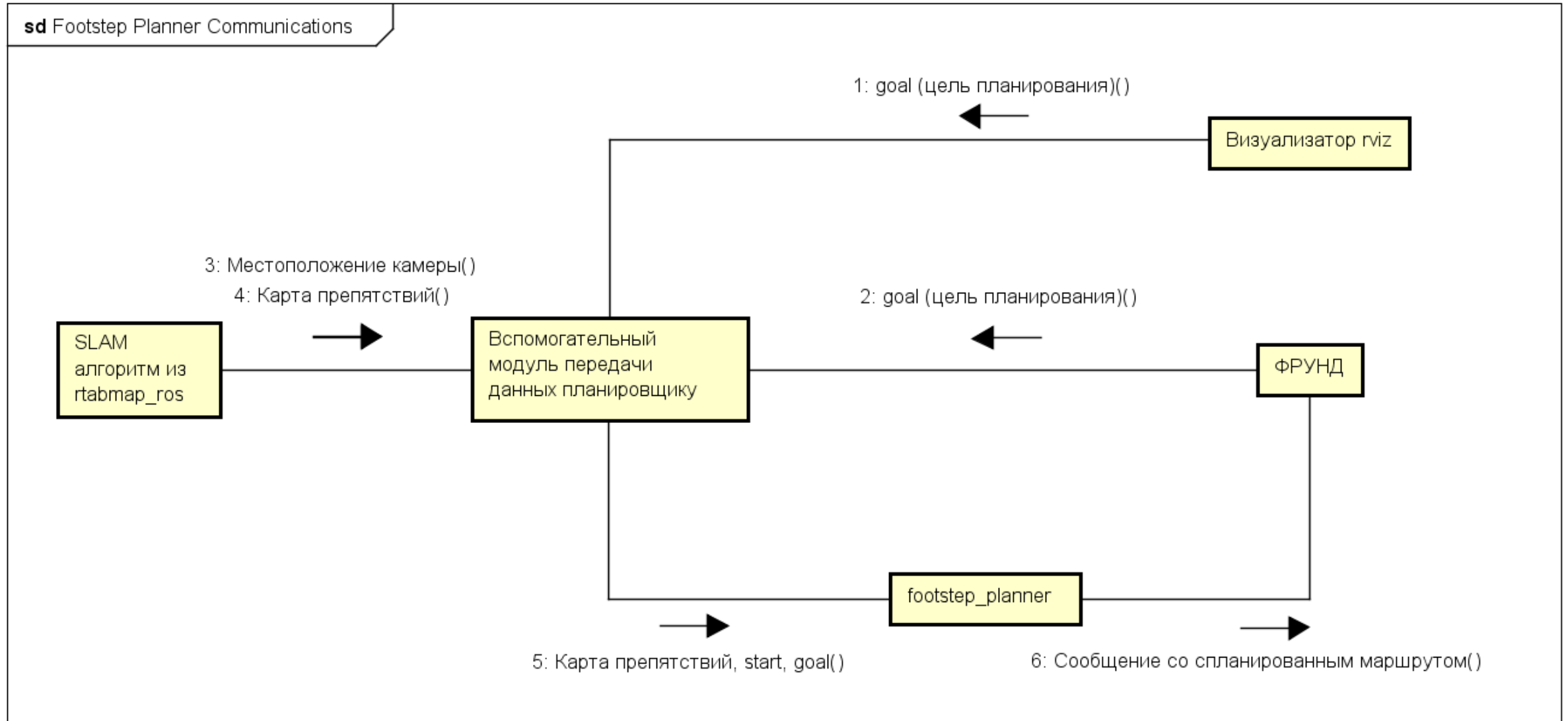


# Принцип планирования

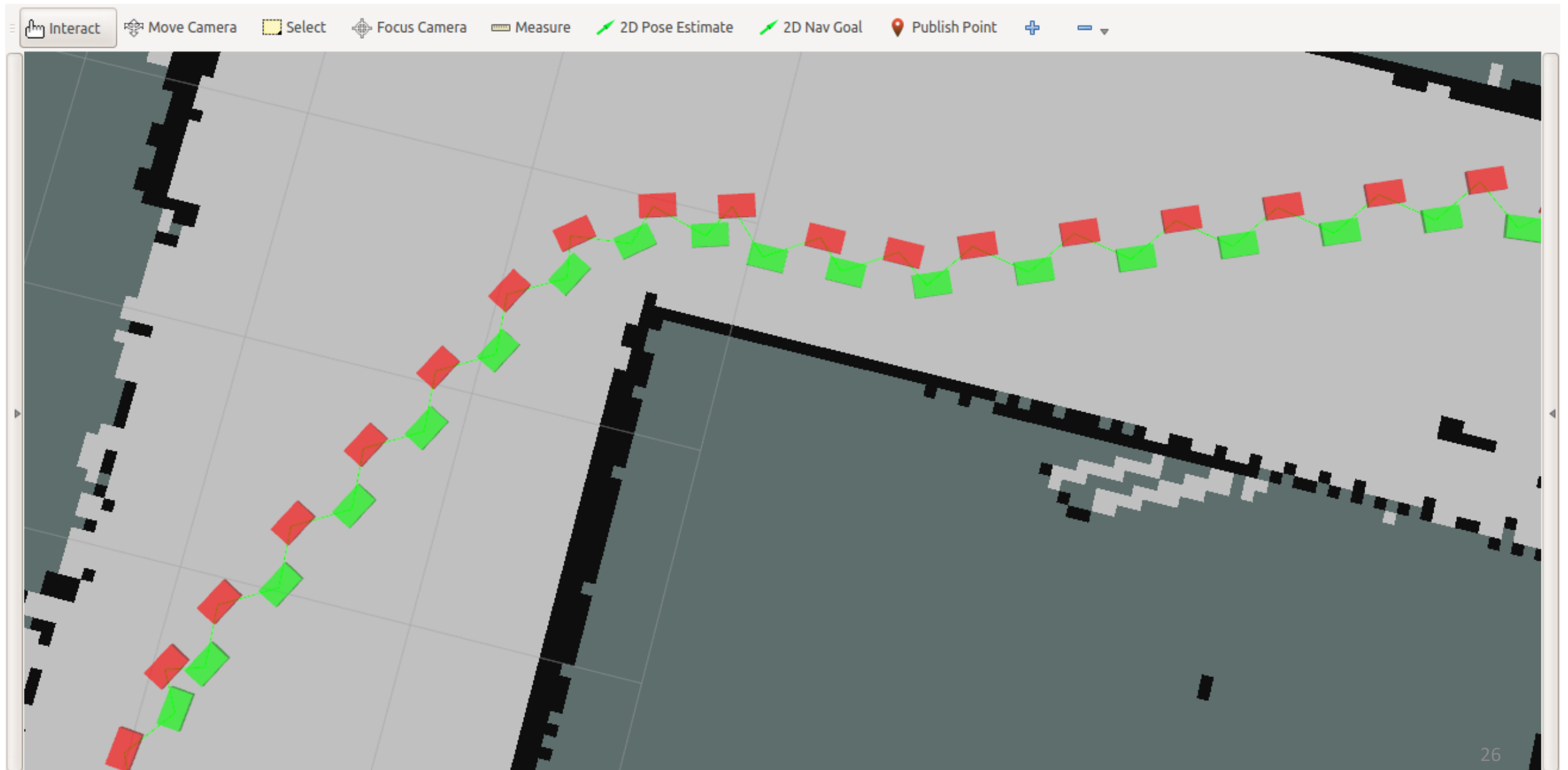


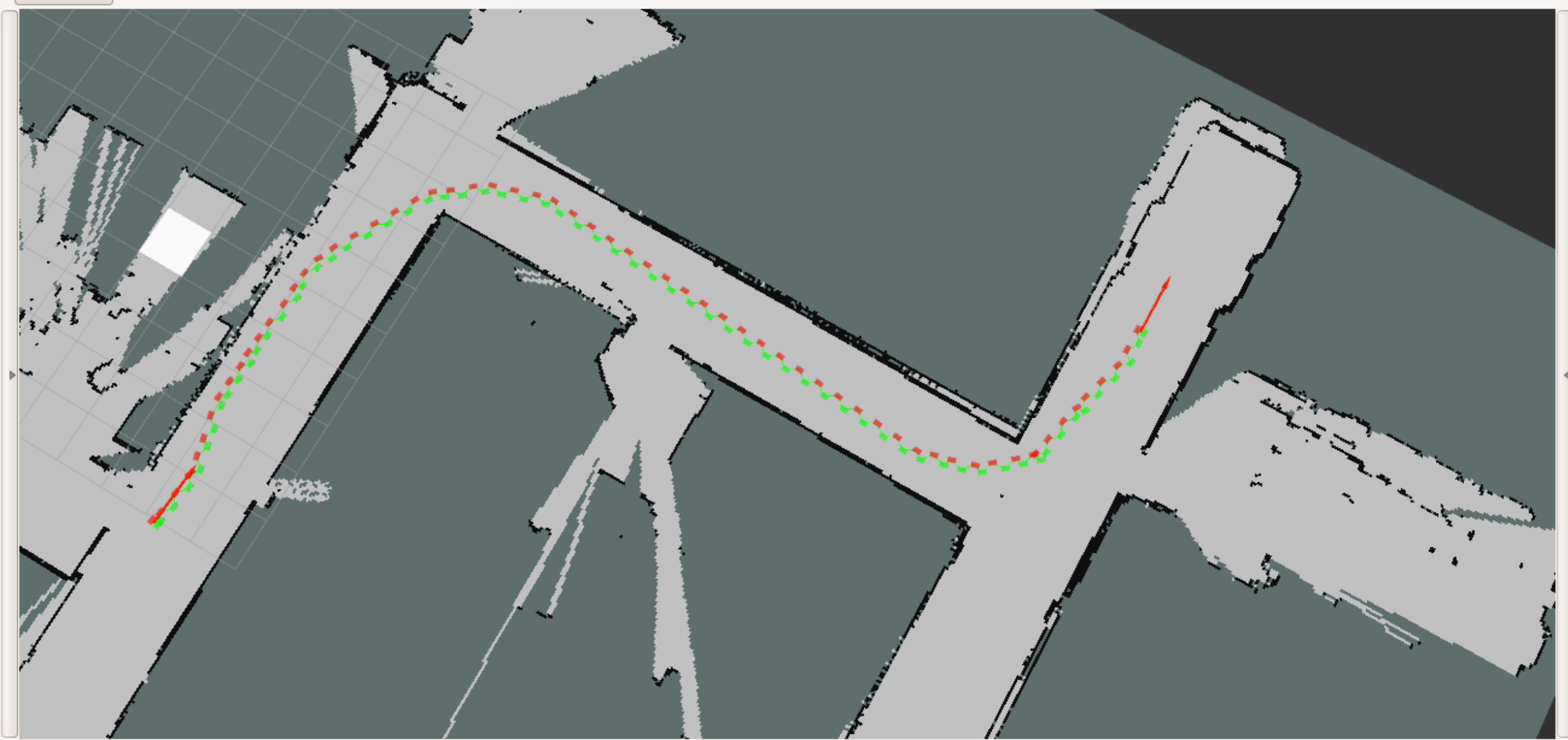


# Архитектура модуля



# Результаты планирования

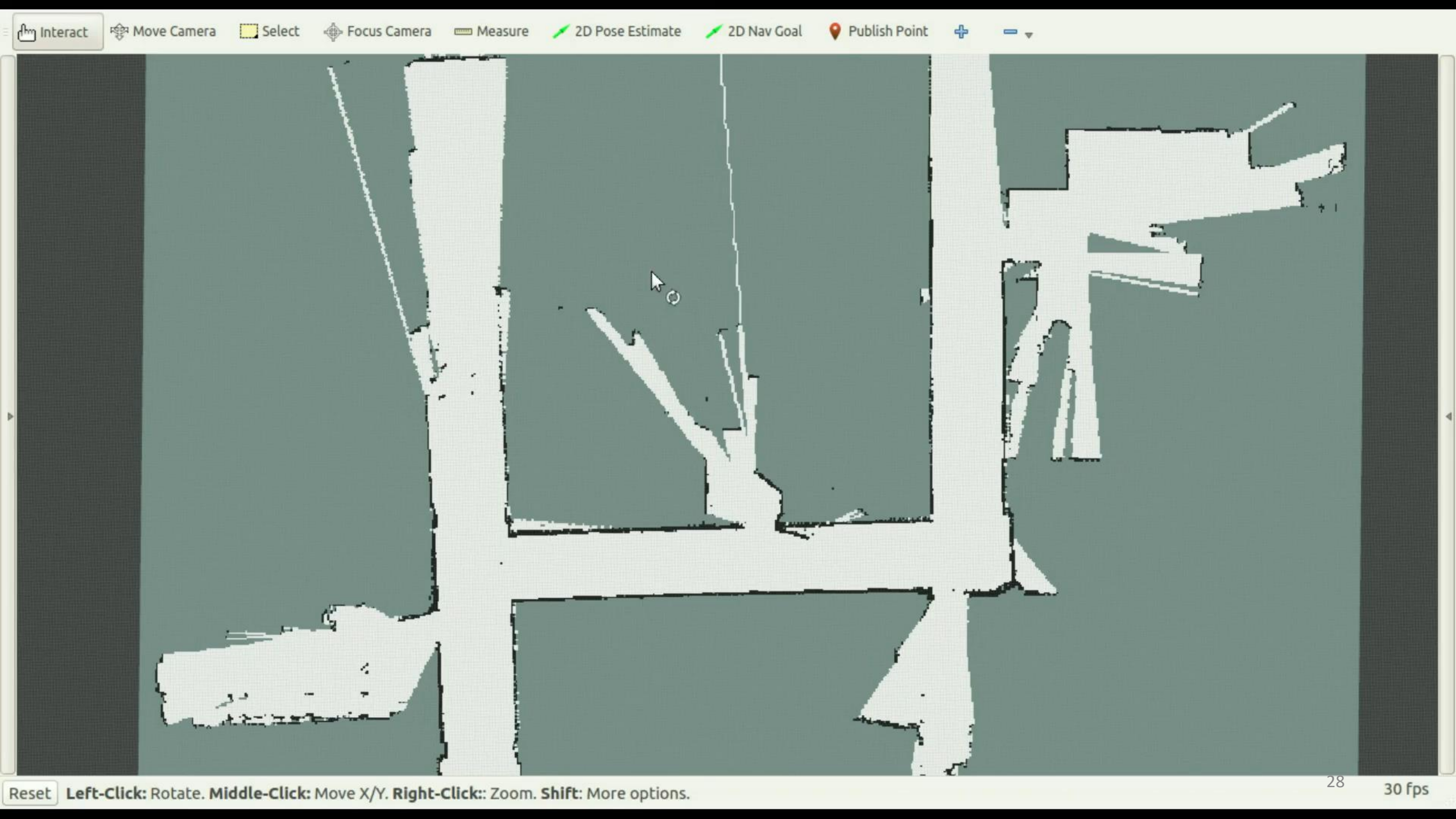


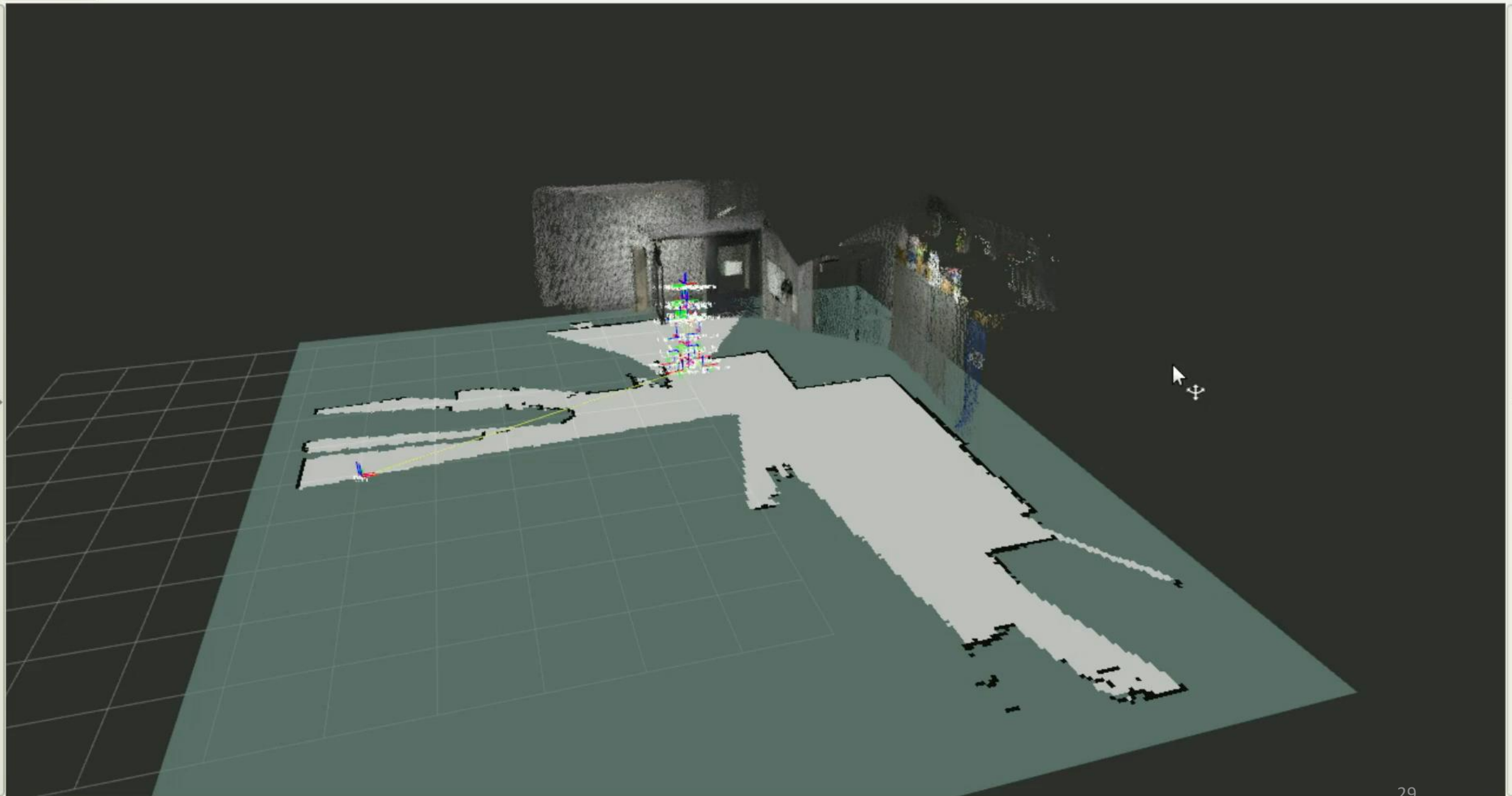


Time

ROS Time: 1368730171.79 ROS Elapsed: 168.31 Wall Time: 1497884829.69 Wall Elapsed: 168.29 ☐ Experimental 27

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options. 30 fps

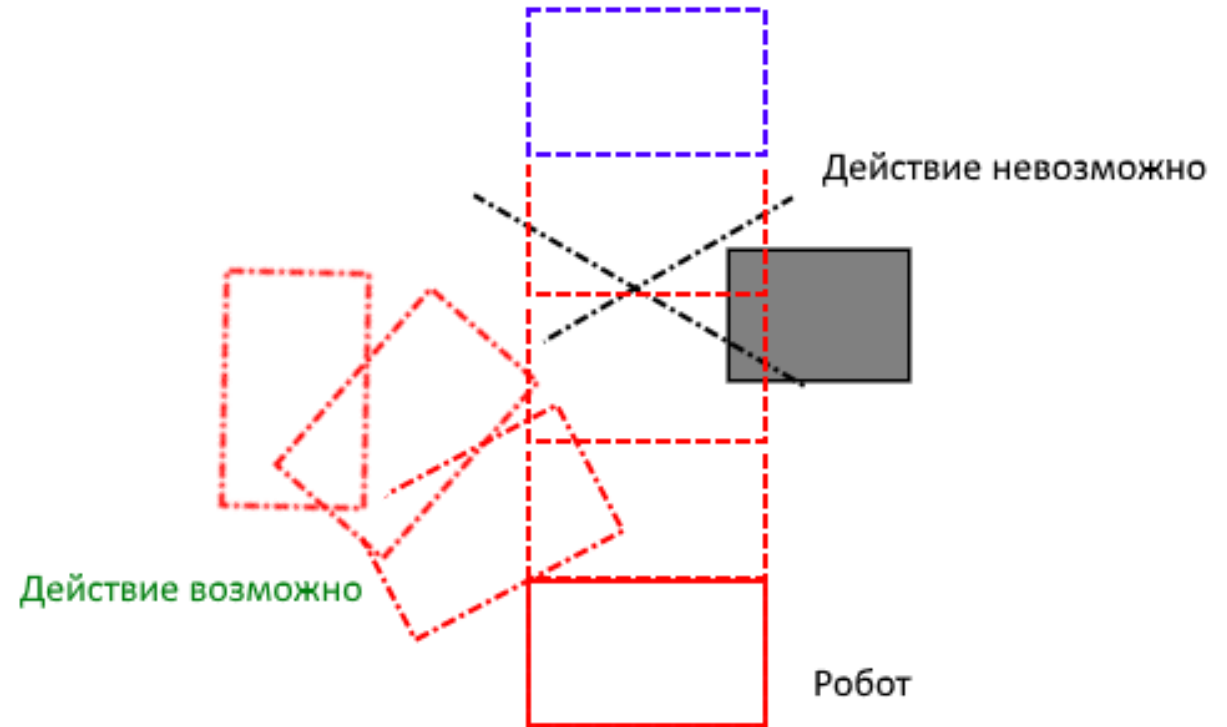
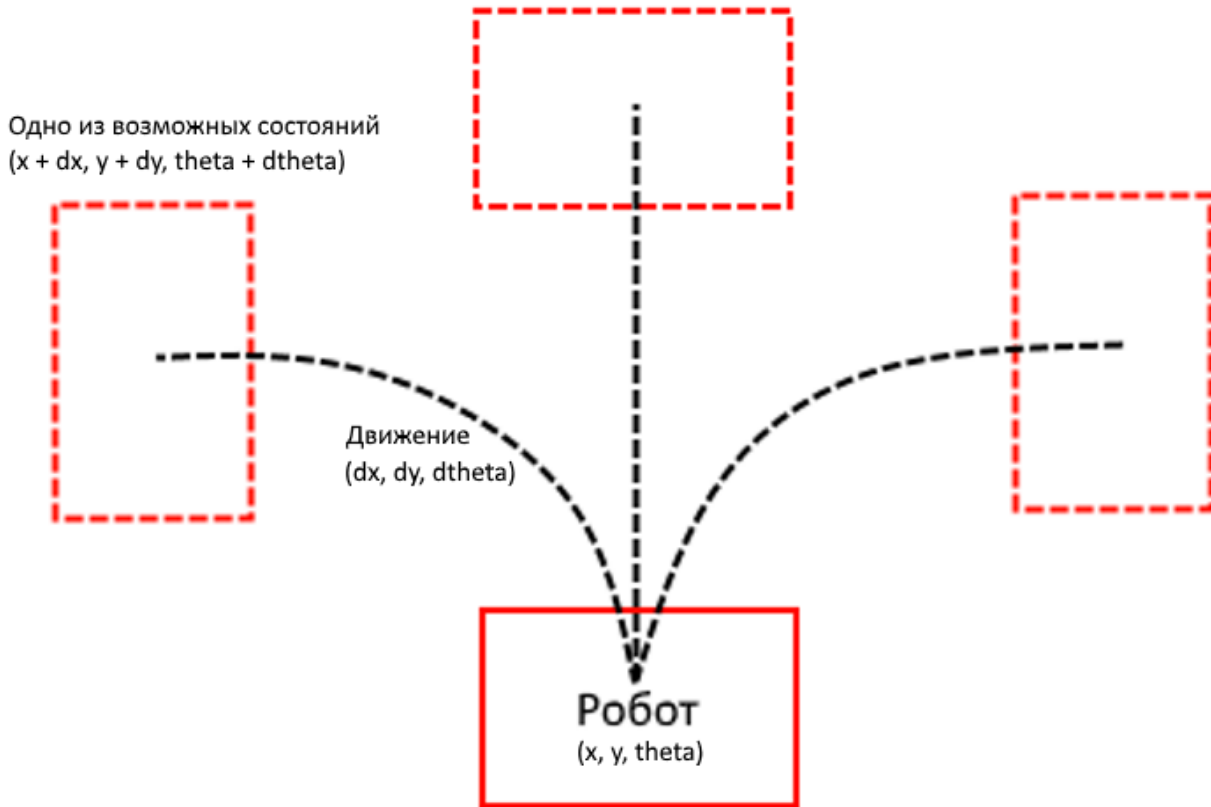




# Реализация модуля планирования маршрута

Вариант 2. Планирование траектории в виде линии (набора точек)

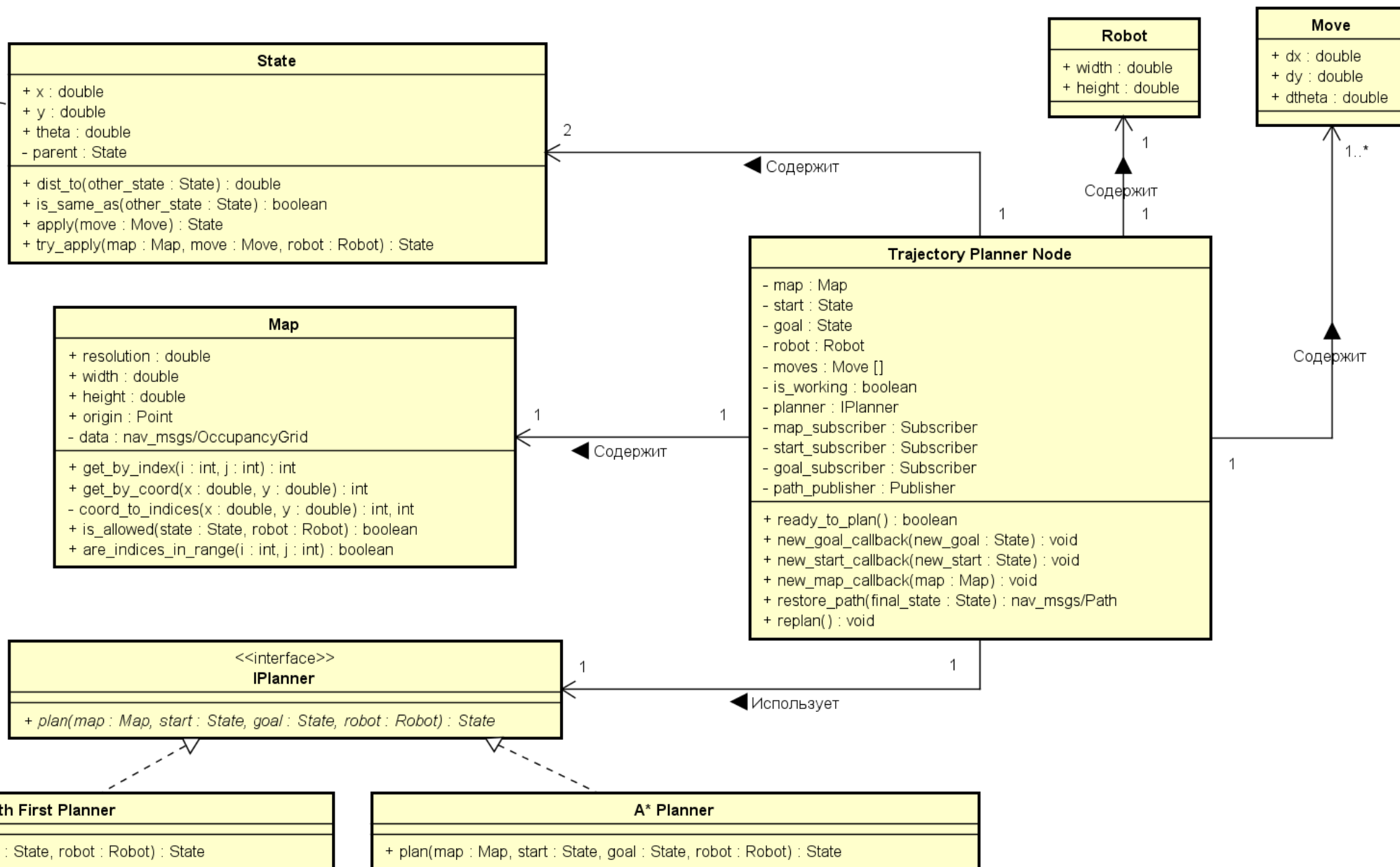
# Принцип планирования



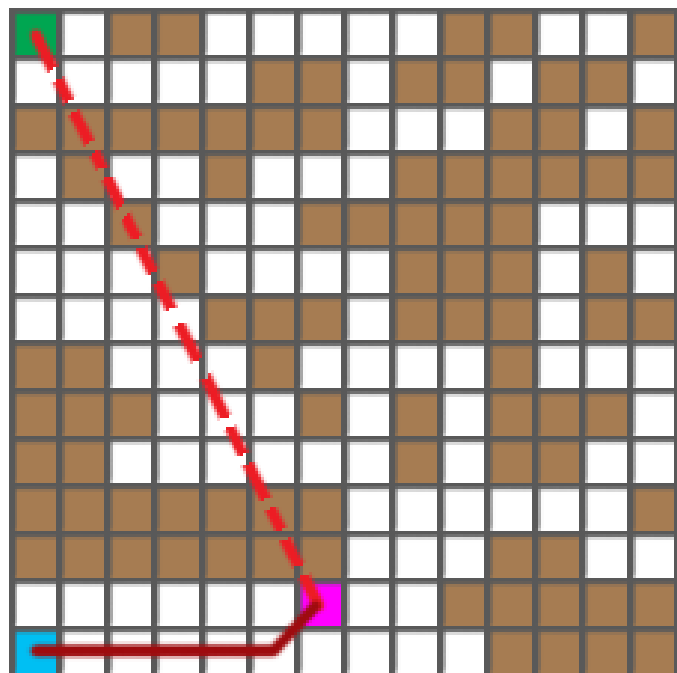
# Архитектура модуля планирования траектории



Здесь могут быть дополнительные поля, необходимые для планировщика



# Алгоритм планирования (A\*)



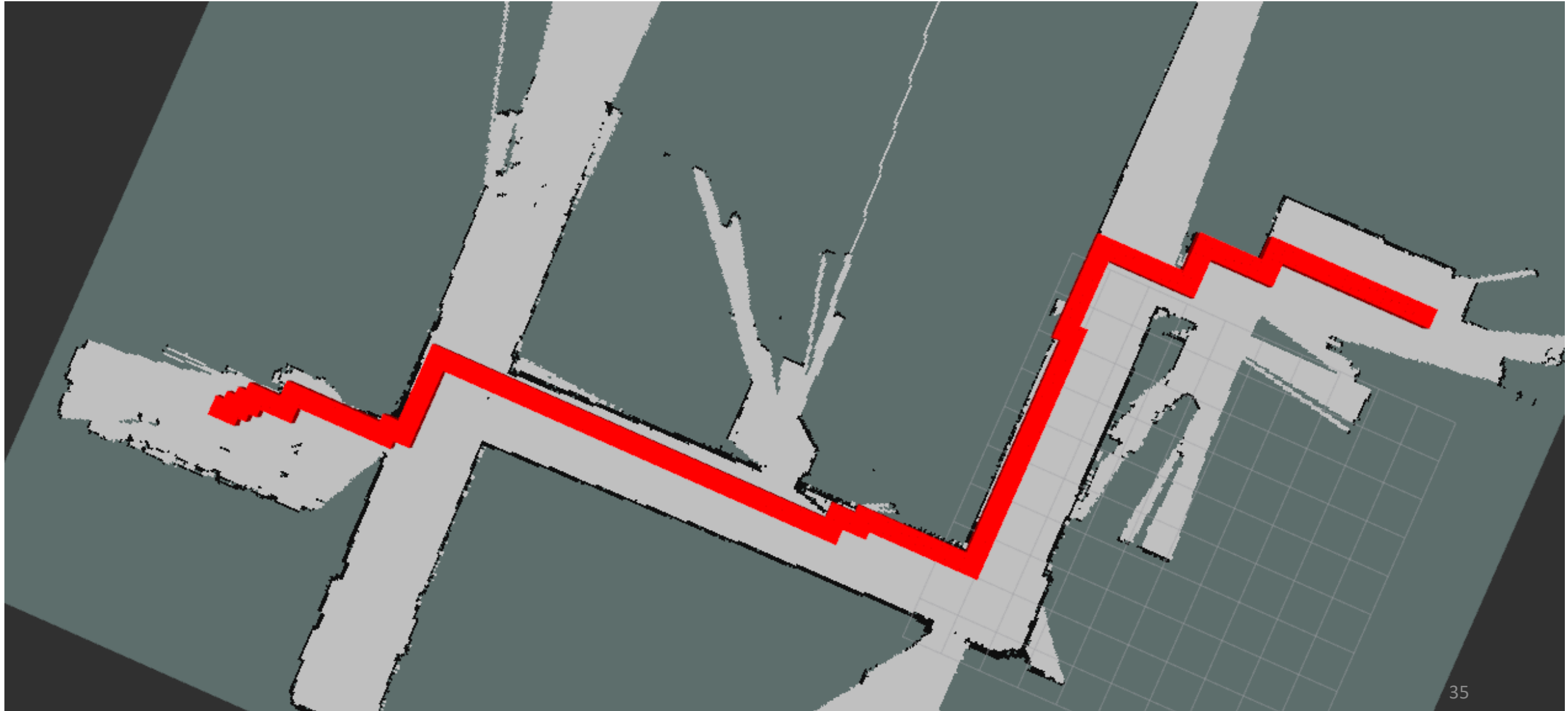
```
struct node {
    node *parent;
    int x, y;
    float f, g, h;
}

// A*
initialize the open list
initialize the closed list
put the starting node on the open list (you can leave its f at zero)

while the open list is not empty
    find the node with the least f on the open list, call it "q"
    pop q off the open list
    generate q's 8 successors and set their parents to q
    for each successor
        if successor is the goal, stop the search
        successor.g = q.g + distance between successor and q
        successor.h = distance from goal to successor
        successor.f = successor.g + successor.h

        if a node with the same position as successor is in the OPEN list \
            which has a lower f than successor, skip this successor
        if a node with the same position as successor is in the CLOSED list \
            which has a lower f than successor, skip this successor
        otherwise, add the node to the open list
    end
    push q on the closed list
end
```

# Результаты планирования



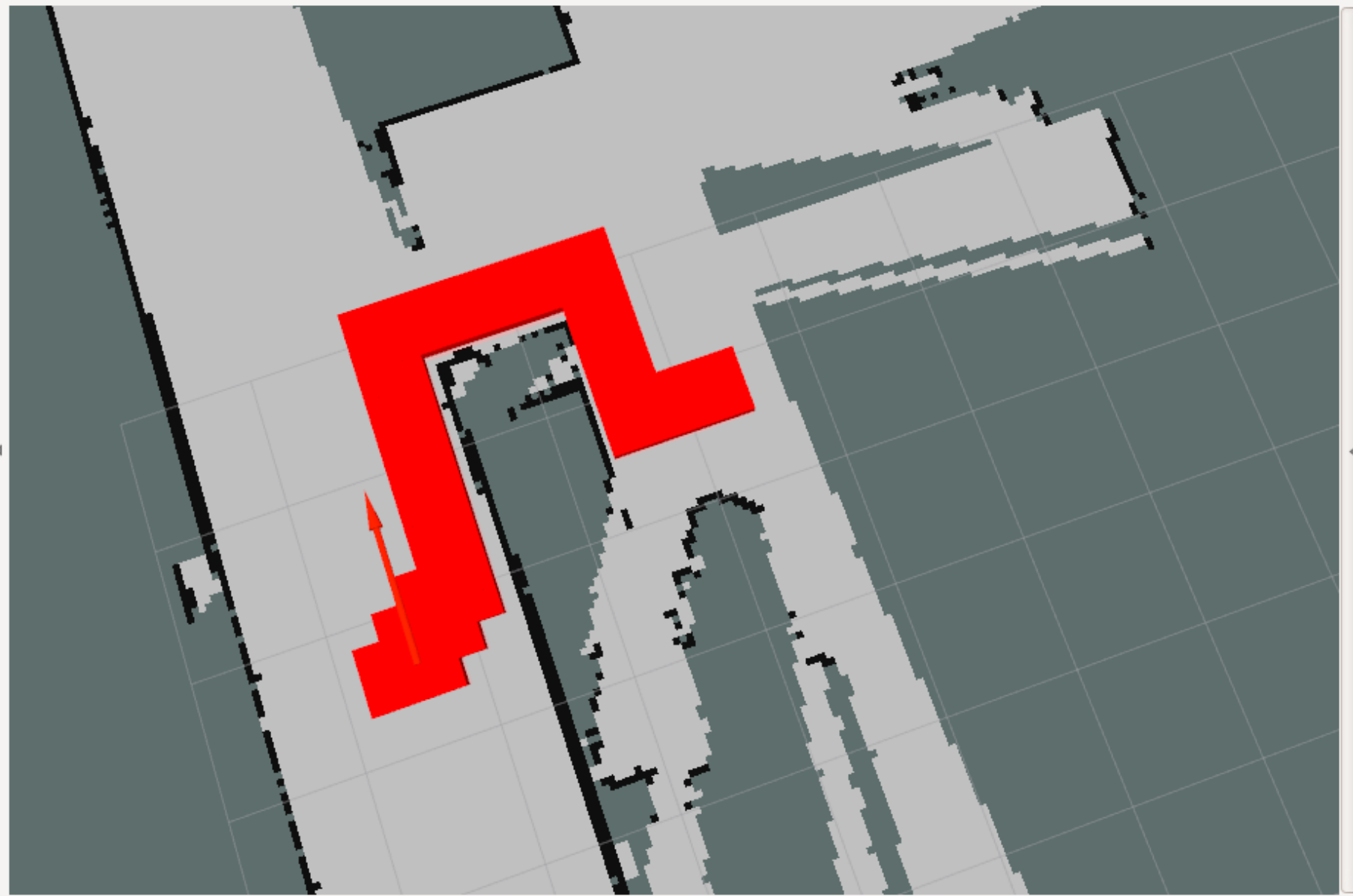
## Displays

- Global Options
  - Fixed Frame
  - Background Color
  - Frame Rate
- Global Status: Warn
  - Fixed Frame
- Grid ☒
- Pose ☒
- Map ☒
- Pose ☒
- MarkerArray ☒

map  
48; 48; 48  
30  
No tf data. Actual error...

**Pose**  
Displays a geometry\_msgs::PoseStamped message.  
[More Information.](#)

Add Duplicate Remove Rename



Time

ROS Time: 1497888572.80 ROS Elapsed: 2142.73 Wall Time: 1497888572.83 Wall Elapsed: 2142.73

☐ Experimental 36

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.

30 fps

Interact

Move Camera

Select

Focus Camera

Measure

2D Pose Estimate

2D Nav Goal

Publish Point

+

-

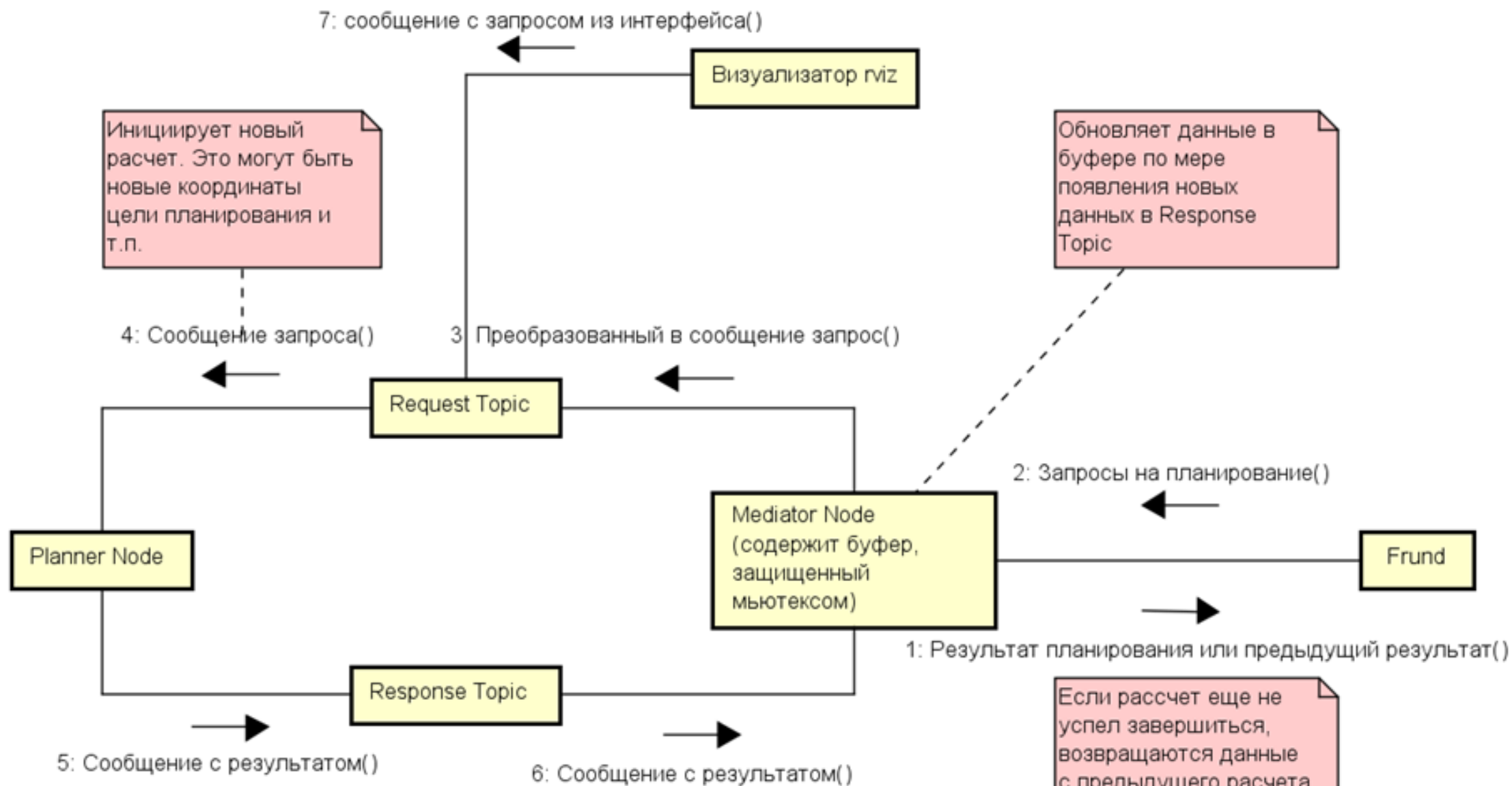


Reset

37

30 fps

# Система взаимодействия модулей компьютерного зрения и ФРУНДа



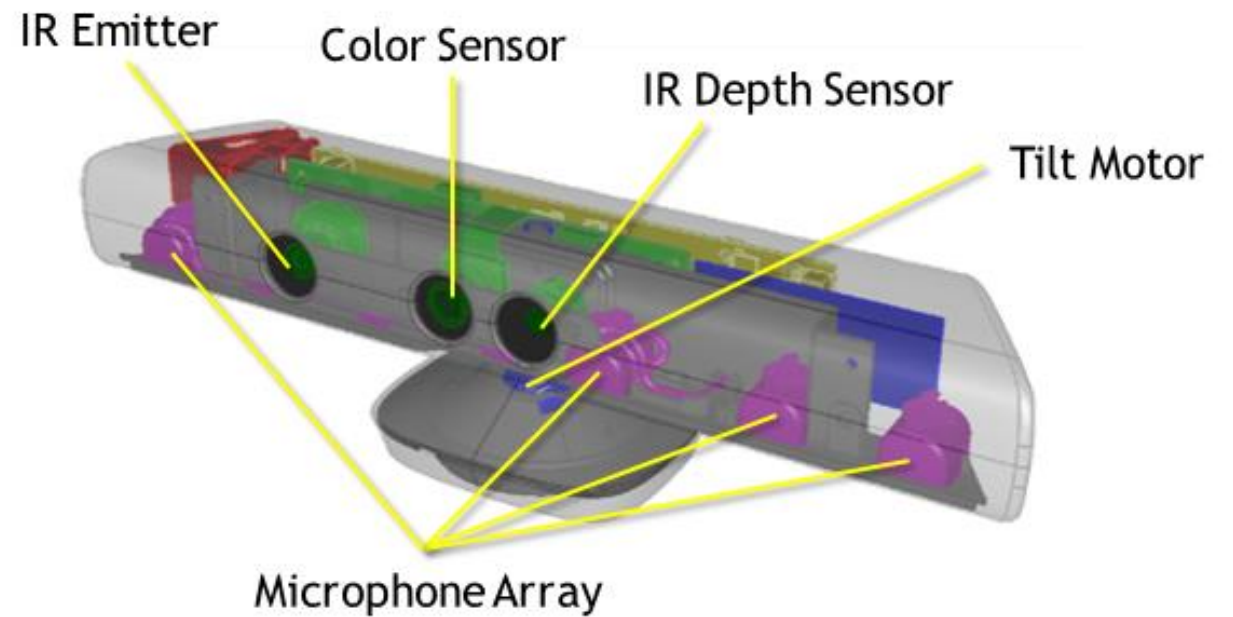
# Дальнейшие работы

- Доработка коммуникации между модулями системы и ФРУНДом, доработка модуля обработки полученной информации и учета ее при планировании на стороне ФРУНДа.
- Доработка планировщика траектории и публикация его на Wiki ROS.
- Эксперименты с планированием движений робота с использованием пакетов Move It и Drake. И работа над учетом окружения при планировании во ФРУНДе.



# Вопросы

# Характеристики Kinect'a



Kinect	Array Specifications
Viewing angle	43° vertical by 57° horizontal field of view
Vertical tilt range	±27°
Frame rate (depth and color stream)	30 frames per second (FPS)
Audio format	16-kHz, 24-bit mono pulse code modulation (PCM)
Audio input characteristics	A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing including acoustic echo cancellation and noise suppression
Accelerometer characteristics	A 2G/4G/8G accelerometer configured for the 2G range, with a 1° accuracy upper limit.

# Качественная оценка точности результатов SLAM алгоритма из rtabmap\_ros



# Системы координат

