XII международная конференция
CEE-SECR / РАЗРАБОТКА ПО
28 - 29 октября, Москва

Библиотека для разработки алгоритмов SLAM в ROS

Дмитрий Карташов
Академический Университет

Введение

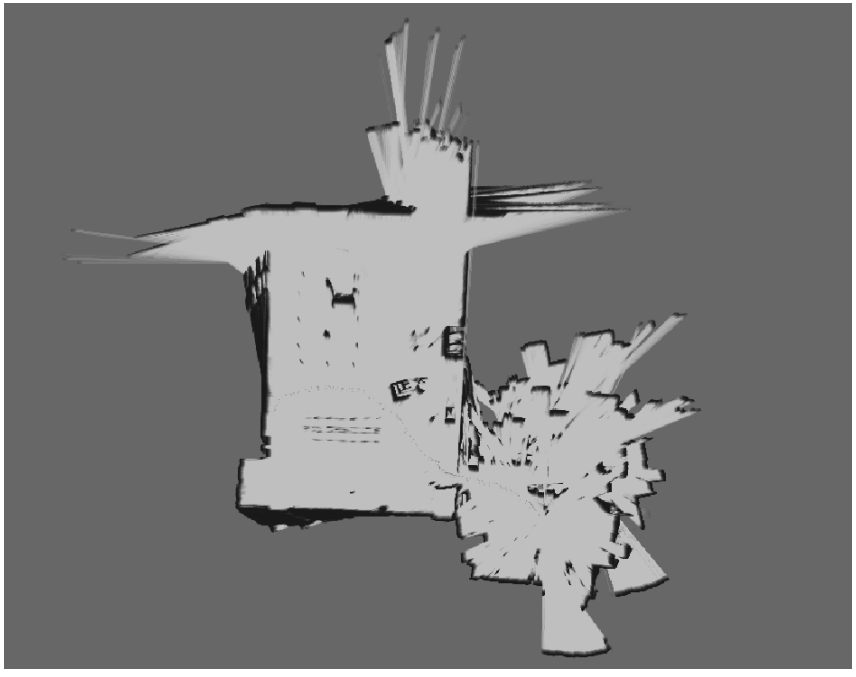
SLAM (Simultaneous Localization And Mapping) – задача построения карты пространства, структура которого заранее неизвестна, и одновременного определения позиции робота в нем.

В настоящее время:

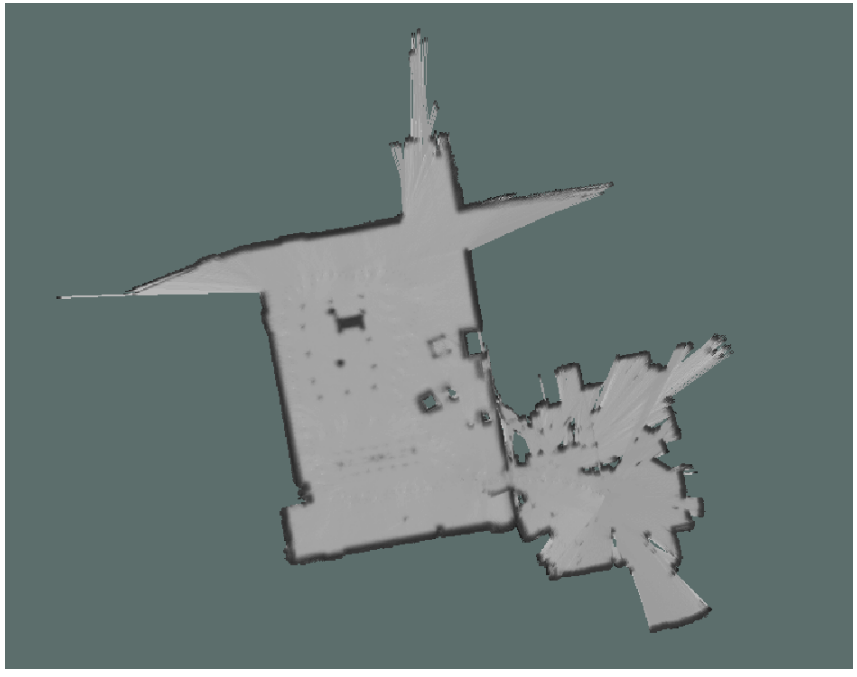
- задача **локализации** решается с приемлемой точностью по данным сенсоров и имеющейся достоверной карте;
- задача **построения карты** решается при известной точной траектории робота и данным сенсоров;
- общего подхода, решающего эти задачи одновременно с заданной точностью и вне зависимости от особенностей окружения, на данный момент **не существует**.

Применение SLAM-методов

Пример карт, полученных с использованием только “сырых” данных сенсоров (a), и при применении одного из SLAM-методов (b):



(a) “Сырые” данные



(b) SLAM

Общий подход к решению задачи SLAM

Вероятностный метод, основанный на теореме Байеса:

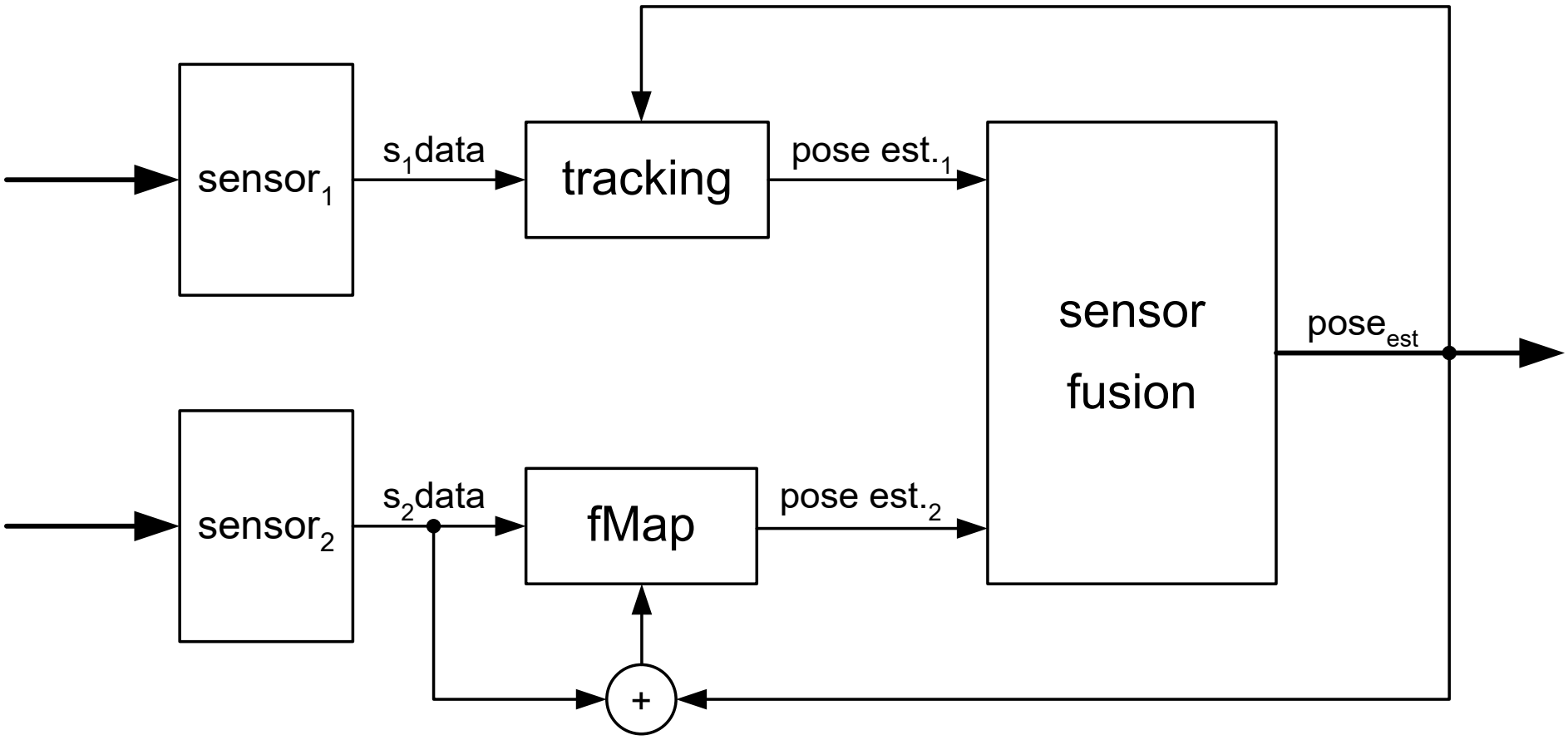
1. Оценка позиции робота: цикл **предсказание** (данные одометрии) – **коррекция** (внешние наблюдения).
2. Обновление карты, используя внешние наблюдения и скорректированную позицию.

Классы SLAM-алгоритмов

Методы решения задачи SLAM можно объединить в классы:

- по типу входных данных (используемых роботом сенсоров):
  - лазерный сканер и одометрия;
  - камера и одометрия (визуальная одометрия);
  - стереокамера или RGB-D камера;
  - сенсор магнитного поля;
- по типу выходной карты:
  - сетка занятости (occupancy grid);
  - граф (pose graph);
  - комбинированная.

Обобщенная схема SLAM-алгоритмов



2D-SLAM: терминология

Лазерный скан (laser scan) — набор измерений лазерного сканера (точек), каждое из которых описывается расстоянием до препятствия и углом (направлением луча лазера).

Одометрия (odometry) — данные датчиков перемещения, используемые для оценки изменения позиции робота с течением времени.

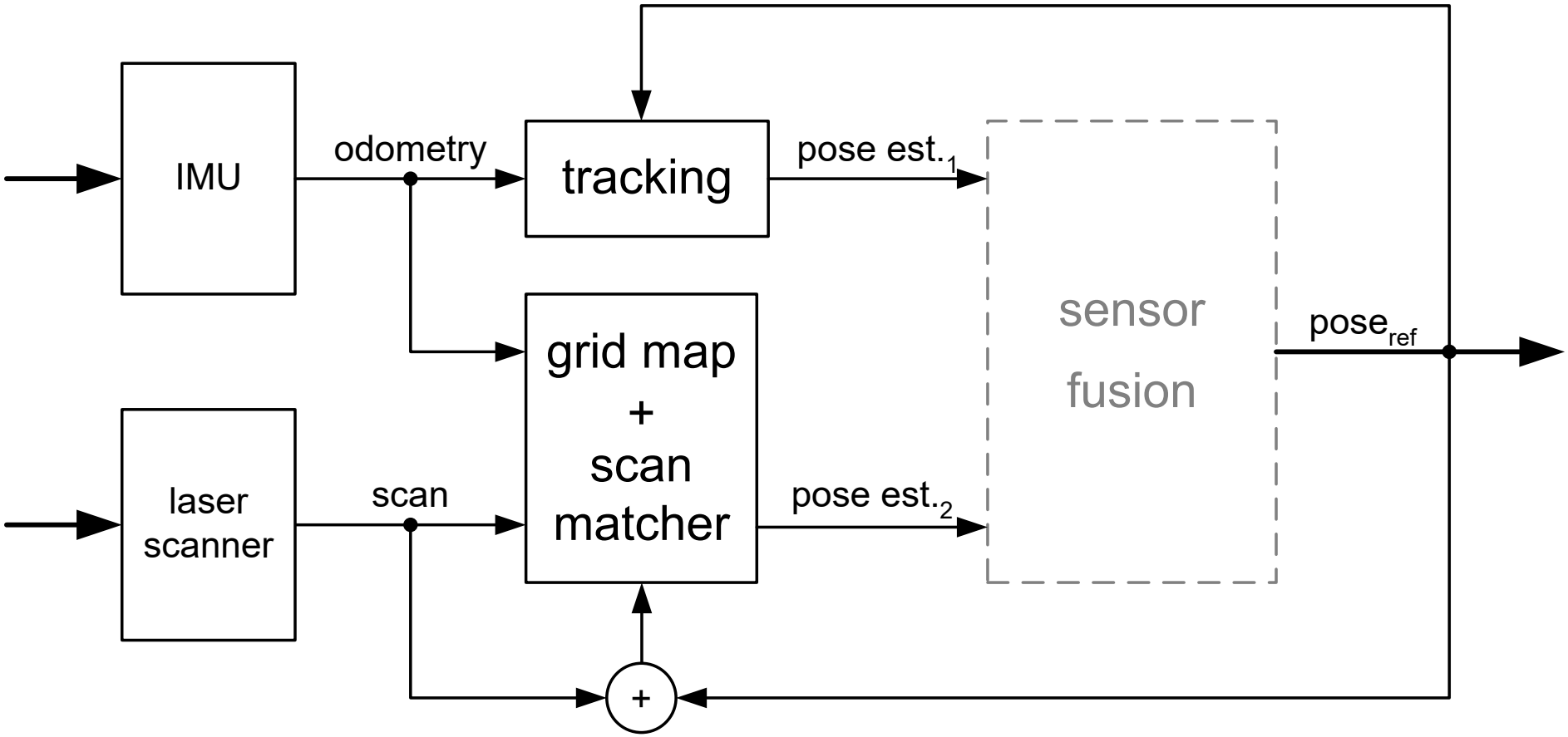
Сетка занятости (occupancy grid) — представление карты окружающего пространства в виде равномерной сетки, каждая ячейка которой хранит вероятность наличия препятствия в этом месте пространства.

2D-SLAM: терминология (продолжение)

Скан-матчер (scan matcher) — алгоритм сопоставления двух лазерных сканов (либо лазерного скана и карты), вычисляющий трансформацию (перемещение и поворот) между этими сканами.

Фильтр частиц (particle filter) — способ оценки параметров системы (А), которые нельзя измерить напрямую, используя измерения других параметров (Б), связанных с первыми. Фильтр создает множество гипотез (частиц) о текущем значении параметров А, после чего итеративно отсеивает недостоверные гипотезы, основываясь на измерениях параметров Б.

Схема SLAM, использующего лазерный сканер



Примеры: **tinySLAM**, **GMapping**

TinySLAM

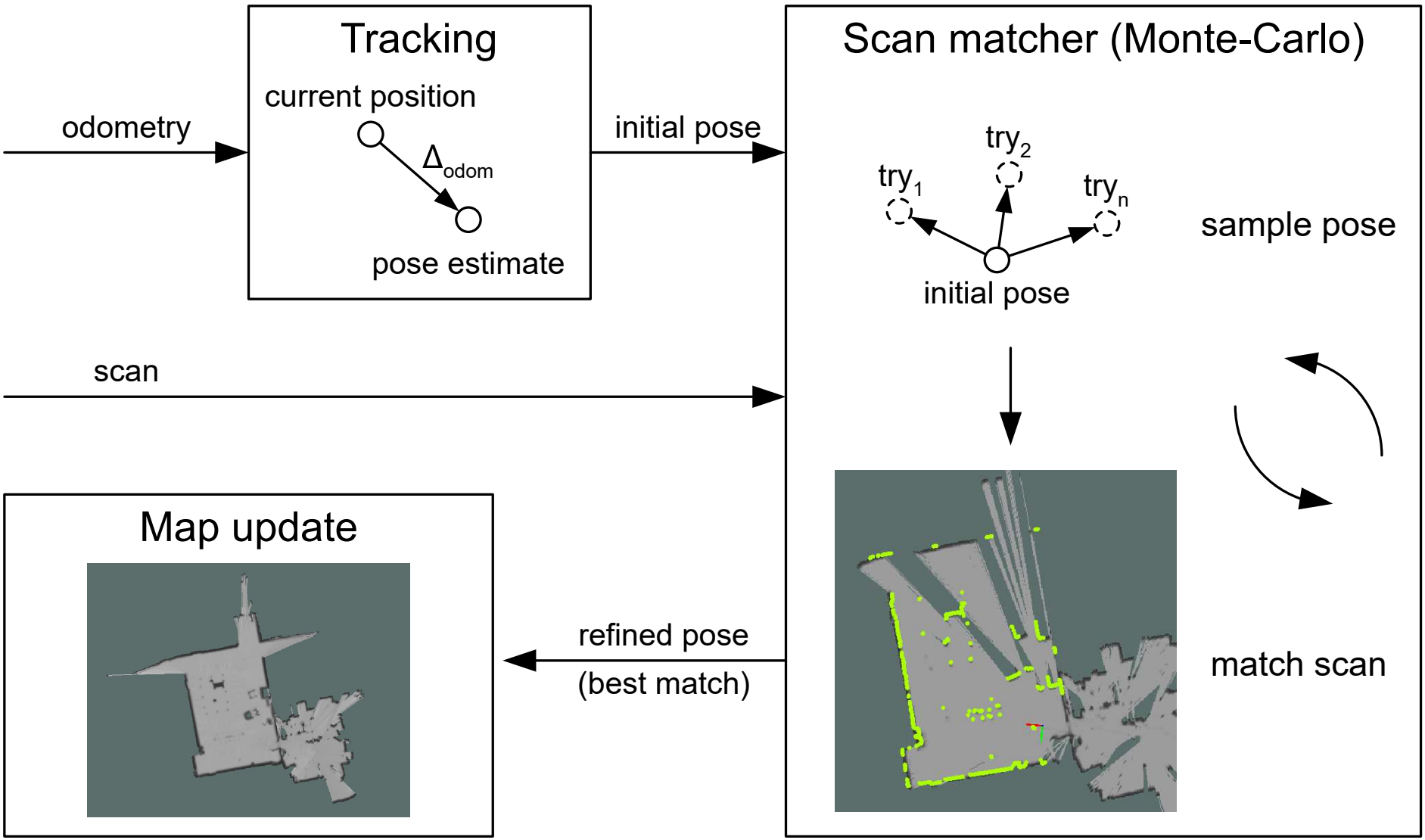
TinySLAM[1] — один из самых простых для реализации и понимания SLAM-методов.

Особенности:

- входные данные – одометрия и лазерный скан;
- структура окружающего пространства представляется в виде сетки занятости;
- положение робота задается 2D-координатами и направлением;
- поддерживает ровно одну гипотезу о состоянии мира (карта + позиция робота).

[1] B. Steux, O. E. Hamzaoui. “tinySLAM: A SLAM algorithm in less than 200 lines C-language program”, 2010.

Алгоритм TinySLAM



GMapping

GMapping[2] — один из наиболее известных 2D SLAM методов.

Особенности:

- основан на фильтре частиц Рао-Блэквелла (RBPF);
- поддерживает одновременно несколько гипотез о состоянии мира;
- представление окружающего пространства – сетка занятости, входные данные – лазерный скан и одометрия.

[2] G. Grisetti, C. Stachniss, W. Burgard. "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling", 2005.

GMapping: детали

- каждая частица хранит отдельную копию состояния мира (карта и положение робота), частица с наибольшим весом считается актуальным состоянием мира;
- скан-матчер является вариацией градиентного спуска: на каждой итерации тестируются несколько дискретных направлений смещения робота, после чего направление с наилучшим значением функции корреляции (matching score) становится исходной позицией для следующей итерации;
- matching score всего скана вычисляется как сумма очков каждой точки скана: чем ближе точка скана в предполагаемой позиции робота к препятсвию на карте, тем больше очков она получает;
- после обработки скана вес каждой частицы пересчитывается пропорционально максимальному значению matching score, полученному в этой частице.

Разработка новых SLAM-методов

Проблема

- Обычно новый алгоритм вносит изменения только в один или несколько модулей существующего SLAM-метода.
- Реализация нового SLAM-метода “с нуля” усложняет его дальнейшую модификацию.

Решение

- Библиотека, предоставляющая модули для быстрого прототипирования алгоритма SLAM и набор часто используемых методов, объектов и классов.

Преимущества фреймворка

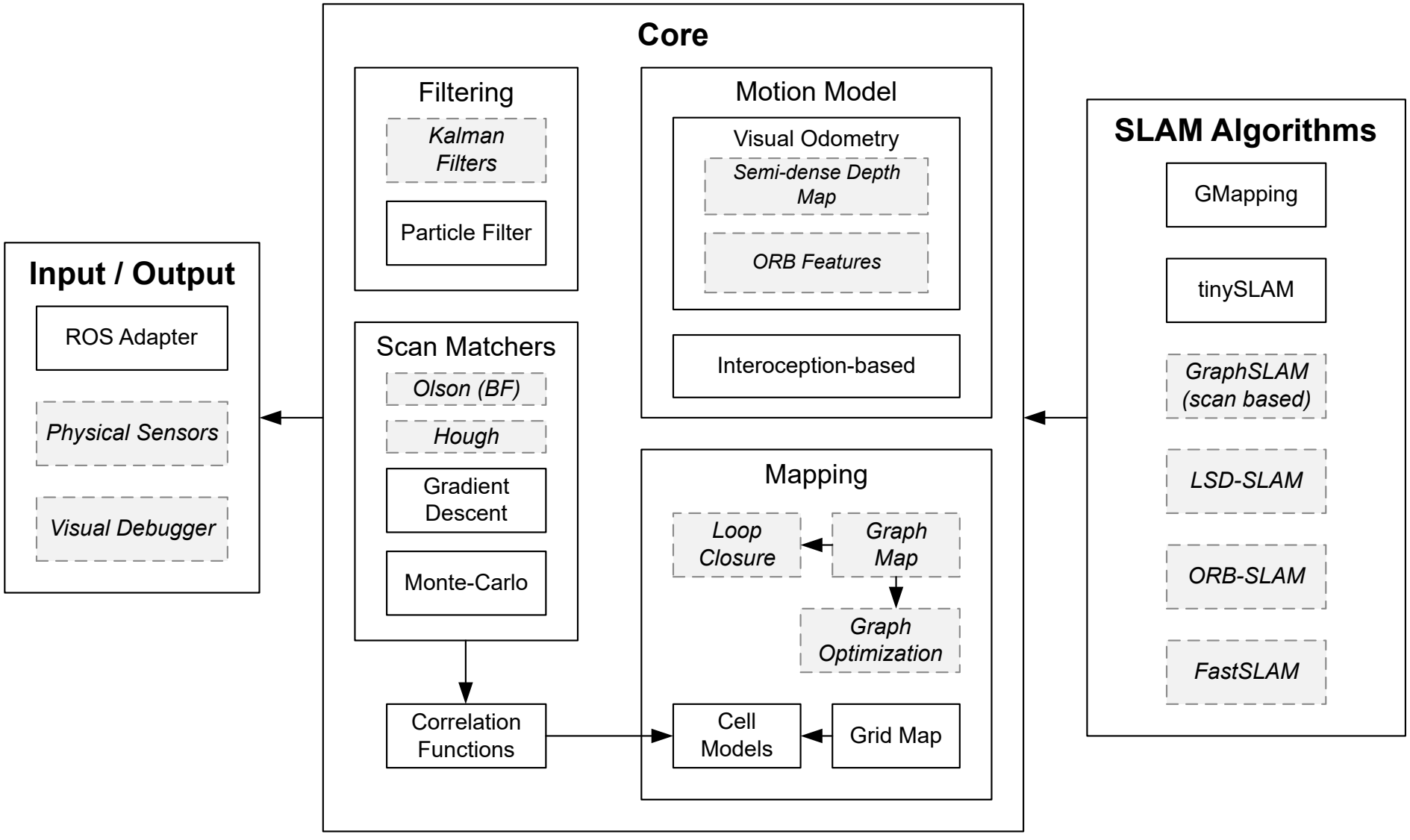
- **ускоряет разработку** новых алгоритмов за счет использования общих компонентов;
- предоставляет возможность объединять части существующих SLAM-методов для **создания нового** более **эффективного** алгоритма;
- **интеграция с ROS** позволяет тестировать алгоитм на популярных датасетах и различных платформах;
- позволяет **выявить детали** и различные ad-hoc особенности, не упомянутые в статье, описывающей конкретный алгоритм.

Существующие решения

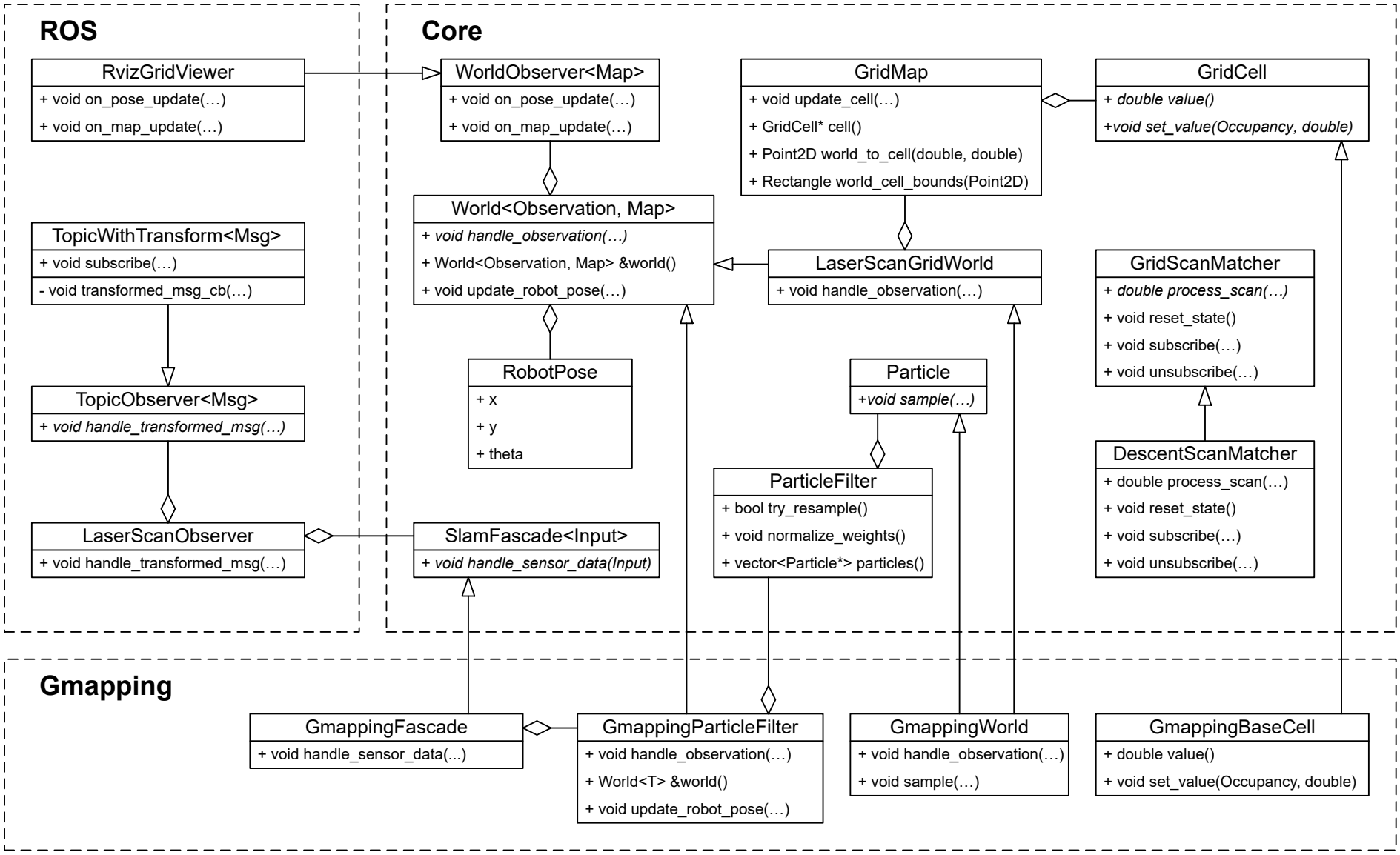
В настоящее время существует несколько наборов библиотек и инструментов, так или иначе связанных с разработкой SLAM-алгоритмов:

- **Manifold Toolkit (MTK)** и библиотека **g2o** упрощают минимизацию нелинейных функций ошибки, что является важной частью графовых SLAM-методов.
- **Mobile Robot Programming Toolkit (MRPT)** предоставляет набор библиотек, полезных для решения практических задач в различных областях робототехники. Он содержит несколько SLAM-методов в виде настраиваемых, но монолитных классов.

Архитектура фреймворка



Пример использования: реализация GMapping



В разработке

- добавить поддержку графовых SLAM-методов: базовые классы и методы оптимизации (ad-hoc или g2o);
- реализовать дополнительные методы сопоставления лазерных сканов;
- добавить поддержку других типов сенсоров (монокулярные или стерео-камеры);
- реализовать дополнительные компоненты и алгоритмы, часто используемые в SLAM (фильтр Калмана и т.п.);
- добавить реализации SLAM-методов, основанных на особых точках (feature-based) (например, FastSLAM).

Заключение

- Библиотека предоставляет базовые компоненты, необходимые для реализации SLAM-методов, основанных на двумерных лазерных сканах.
- Текущая реализация выполнена на языке C++ и совместима с ROS.
- Текущая версия включает в себя реализации таких алгоримов, как tinySLAM и GMapping.
- Разработка библиотеки находится на начальном этапе и множество планируемых компонентов на данный момент в ней отсутствует.
- Текущая версия фреймворка доступна по ссылке: `https://github.com/OSLL/slam-constructor`

Спасибо за внимание!

ros@osll.ru