

Визуальная одометрия

Visual monocular odometry

Андрей Михалев

Алексей Пшеничный

руководитель Дмитрий Карташов

CSC

18 мая 2017 г.

Предметная область

Одометрия

Использование данных о движении приводов для оценки перемещения и построения траектории движения.

Визуальная одометрия

Использование информации из видеопотока, получаемого с камеры-сенсора, для оценки перемещения и построения траектории движения.

SLAM

Simultaneous Localization and Mapping — задача построения карты и одновременной локализации робота на этой карте.

ROS

Robot Operating System — фреймворк для создания приложений робототехники.

ROS:

- Аппаратная абстракция;
- Обработка информации в узлах (nodes);
- Передача информации между нодами путём обмена сообщениями;
- Сообщения объединяются в очереди (topics);
- Открытый исходный код;
- Ориентирована на Unix-подобные системы;

Актуальность и цель

- Задача визуальной одометрии не имеет универсального решения.
- Каждый SLAM-алгоритм предлагает свой способ визуальной оценки одометрии.
- Существующие алгоритмы визуальной одометрии заточены под тип видеопотока (indoor или outdoor).
- Конкретная реализация алгоритма визуальной одометрии зависит от типов сенсоров и их количества.

Идеальное решение:

- Позволит обеспечить более точную навигацию робота.

Цели и задачи

Цель

Найти решение задачи *monocular odometry*, разработав прототип алгоритма визуальной одометрии, использующий данные от одной RGB-камеры.

Задачи:

- Изучить современные алгоритмы визуальной одометрии;
- Выбрать из существующих или реализовать собственный, провести тестирование.

Роли в проекте

Общие задачи

- Изучение существующих решений.
- Интеграция своих реализаций с ROS.

Андрей Михалев

- Создание прототипа алгоритма визуальной одометрии.
- Тестирование прототипа.

Алексей Пшеничный

- Создание тестового окружения.
- Тестирование библиотечных алгоритмов.

Входные данные



Рис. : MIT indoor

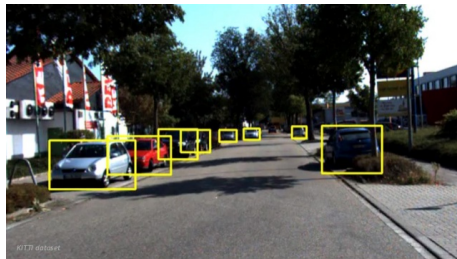
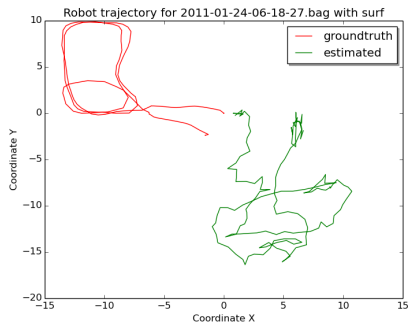
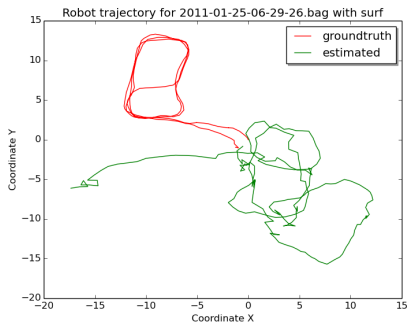


Рис. : KITTI outdoor

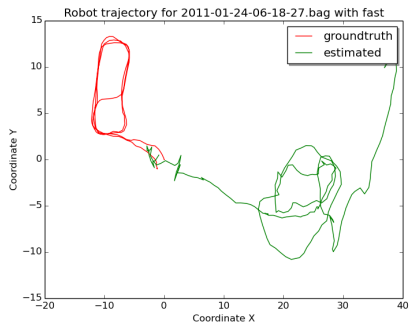
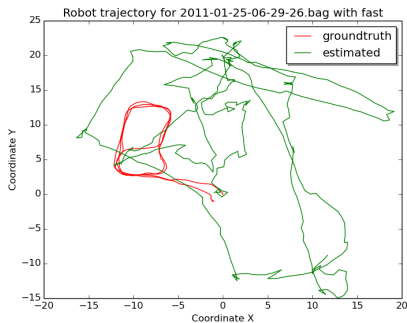
Прототип алгоритма визуальной одометрии

- ❶ Синхронизация показаний одометрии и видеопотка.
 - ❷ Рассматриваются Img_{i-1} и Img_i .
 - ❸ Выполняется оценка визуальной одометрии:
 - OpenCV (Feature matching, feature tracking).
 - LibVisio2.
 - ❹ Показания одометрии и оценка визуальной одометрии записываются в файл.
- По показаниям одометрии и визуальной одометрии строятся траектории движения.

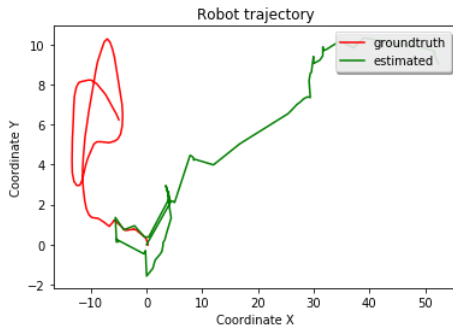
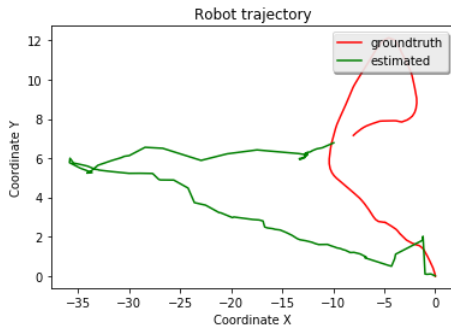
Результаты прототипа для MIT



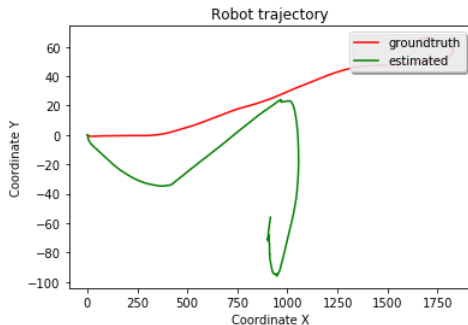
Результаты прототипа для MIT



Результаты LibVisio2



Результаты LibVisio2



- 1 Предложено решение задачи визуальной одометрии.
- 2 Создан прототип алгоритма визуальной одометрии.
- 3 Создано окружение для тестирования библиотечных алгоритмов.
- 4 Нужно более детально изучить и модифицировать этап получения движение робота.

- Сложность современных алгоритмов визуальной одометрии;
- Outdoor и Indoor окружения имеют свои особенности;
- Зашумлённость входных данных мешает созданию универсального алгоритма.
- ROS большой фреймворк с обширным функционалом;

Приобретённые навыки

- Изучение ROS;
- Интеграция кода C++ с ROS;
- Изучение современных методов визуальной одометрии;

Используемые технологии и исходный код

Исходный код и входные данные

- Тестовое окружение:
https://github.com/izhleba/ros_visodom_test
- Прототип алгоритма:
https://github.com/Montura/ros_visual_odometry_base

Технологии

- Robot Operating System (ROS);
- C++11;
- Python;

- Обсуждение, поиск и доработка методов, позволяющих получать преобразования между двумя последовательными кадрами видеопотока.