RWTHAACHEN UNIVERSITY

OpenStreetSLAM: Global Vehicle Localization using OpenStreetMaps

Georgios Floros, Benito van der Zander and Bastian Leibe

Computer Vision Group
RWTH Aachen University, Germany

http://www.vision.rwth-aachen.de
floros@vision.rwth-aachen.de

European Research Council
Established by the European Commission

funded by ERC StG 307432 CV-SUPER

Motivation

Goal: Accurate localization

Applications:

- Autonomous self-driving cars
- Navigation systems

Currently:

- GPS localization

⇨ Not always available (narrow streets)

⇨ Not always accurate

- Velodyne to build own maps

⇨ Expensive

⇨ Huge amount of data to be stored

Alternative:

- Computer Vision

Image source: google.com

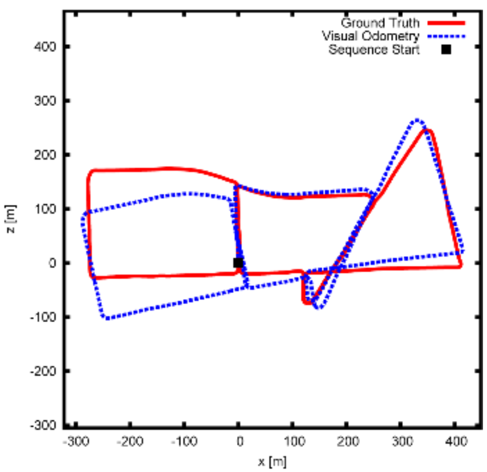
Image source: volvo.com

Related Work

Visual odometry / Visual SLAM

- Monocular e.g. [Nister JFR'06]
- Stereo e.g. [Alcantarilla ICRA'12]

⇨ Accumulated drift makes localization unusable



Extensions

- Bundle adjustment e.g. [Mei IJCV'11]
- Loop closure detection e.g. [Cummins RSS'09]

⇨ Restricted vehicle's motion

Image source: Cummins, RSS'09

Computer Vision Group

Related Work

Image-based localization e.g. [Zamir CVPR'12]

- Geo-tagged image database
- Localization via Image matching

⇨ Image database expensive to build and maintain

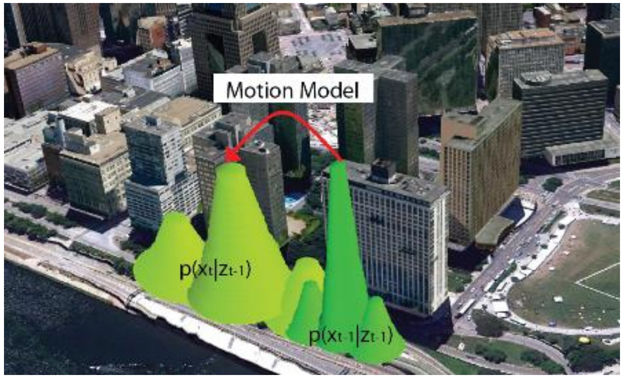


Image source: Zamir, CVPR'12

Contribution

Idea: Use map data (e.g. OpenStreetMaps) to improve localization
Requirement: Registration of vehicle's trajectory on the map

Assumption: Visual Odometry is locally stable and robust
Prerequisite: Rough initial position (uncertainty: ~1km)

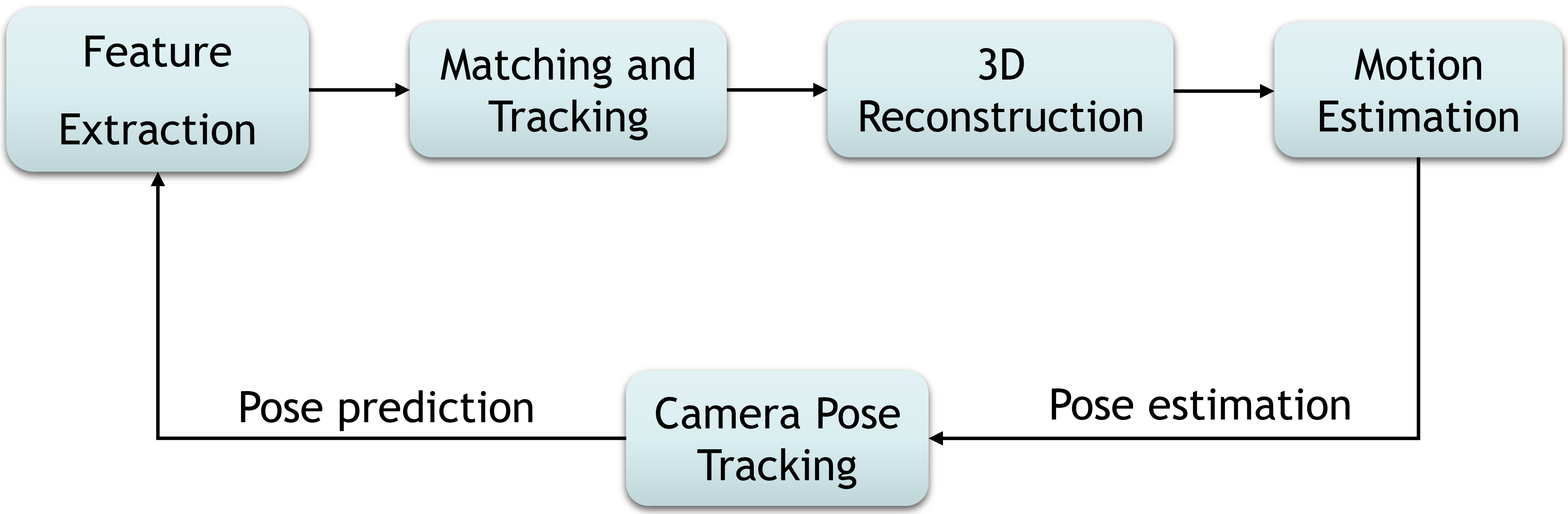
Contributions:
- Automatic global localization on the map
- Local trajectory adjustment according to the map

Advantages:
- No drift accumulation (even when driving without loops)
- Small infrastructure cost

Outline

Visual odometry → Global Localization → Local adjustment

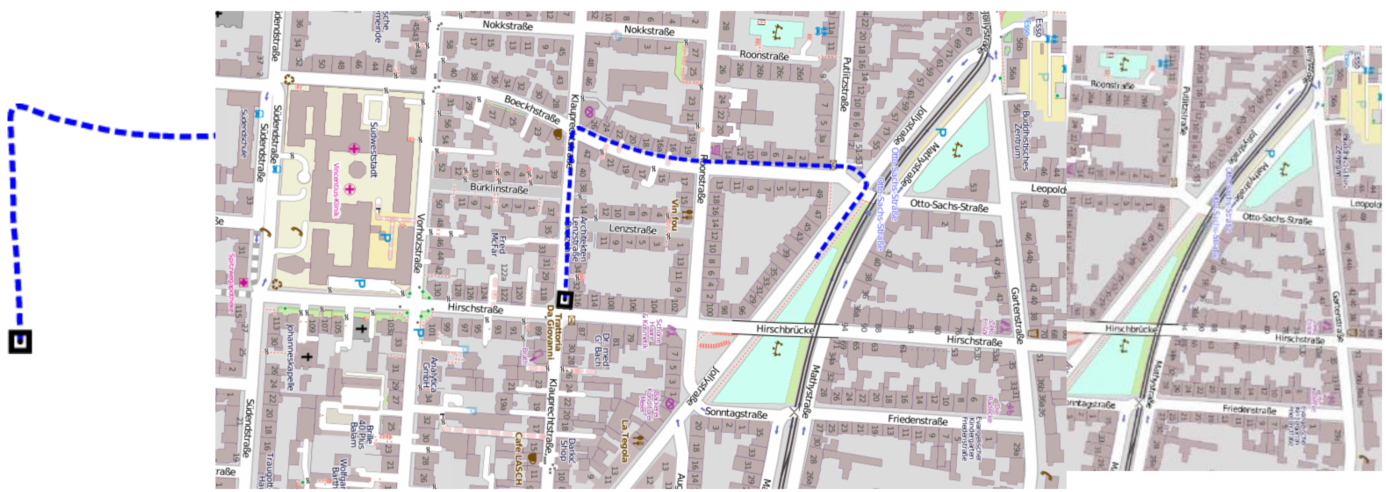
Visual Odometry (VO)



- Visual odometry pipeline based on [Nister JFR06]
- Any VO pipeline could be used

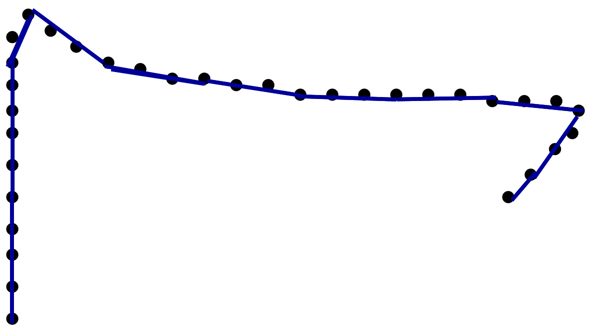
Outline

Visual odometry → Global Localization → Local adjustment



Path to Map – Shape Matching

Camera Path



- Car's trajectory: $C = \{c_0, c_1, \dots, c_N\}$
- Converted to set of line segments
- Query shape: $Q = \{\boldsymbol{q_i}\}$

OSM Map

- OSM Map elements
  - Nodes $n = (lat, lon)$
  - Ways $w = \{n_i\}_{i=1,\dots,k}$
  - Relations r
- Street graph
- Set of line segments
- Template shape: $T = \{\boldsymbol{t_i}\}$

Chamfer Matching

Given:

- Query edge map, $Q = \{\boldsymbol{q_i}\}$
- Template edge map, $T = \{\boldsymbol{t_i}\}$

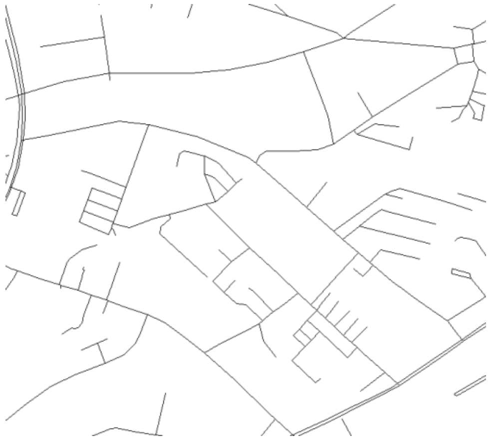
Find

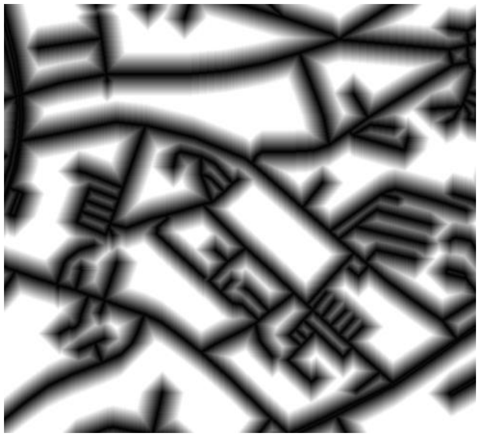
$\boldsymbol{H} = (\theta, t_x, t_y), \boldsymbol{H} \in SE(2)$

which minimizes:

$$d_{CM} = \frac{1}{|W|} \sum_{w_i \in W} \min_{t_j \in T} |w_i - t_j| \, ,$$

where $W = \boldsymbol{H} \cdot Q$



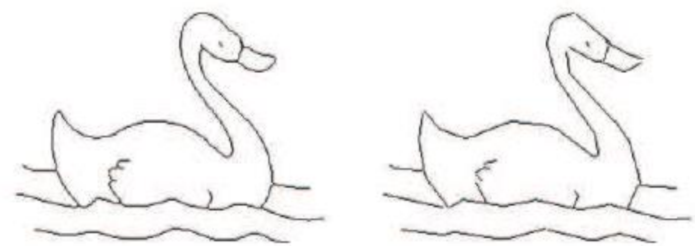


Fast Directional Chamfer Matching

Fast Directional Chamfer Matching (FDCM) [Liu CVPR'10]

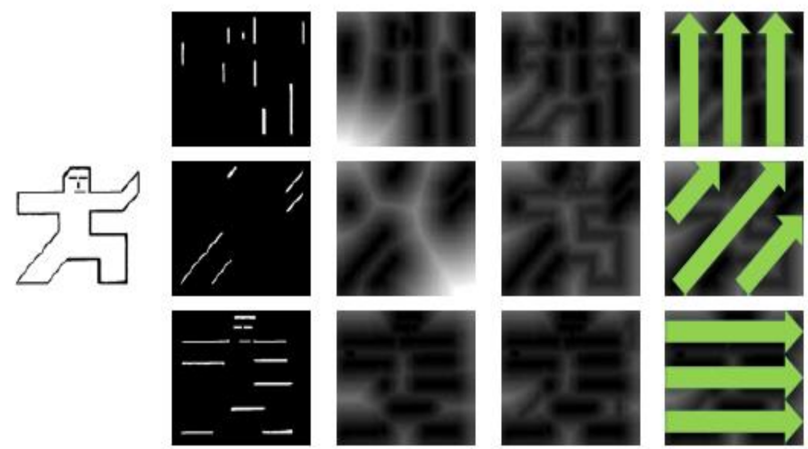
Extensions to Chamfer Matching:

- Use line segments, not edge points
- Incorporates edge information
- Reduce cardinality, m lines $\ll n$ edges



Advantages:

- 45x faster
- Robust to outliers



Suitable for our approach:

- OSM given as line segments
- Path easily converted
- Allows use of large OSM maps

Image source: Liu, CVPR'10

Outline

Visual odometry → Global Localization → Local adjustment

Local path adjustment using MCL

Monte Carlo Localization framework

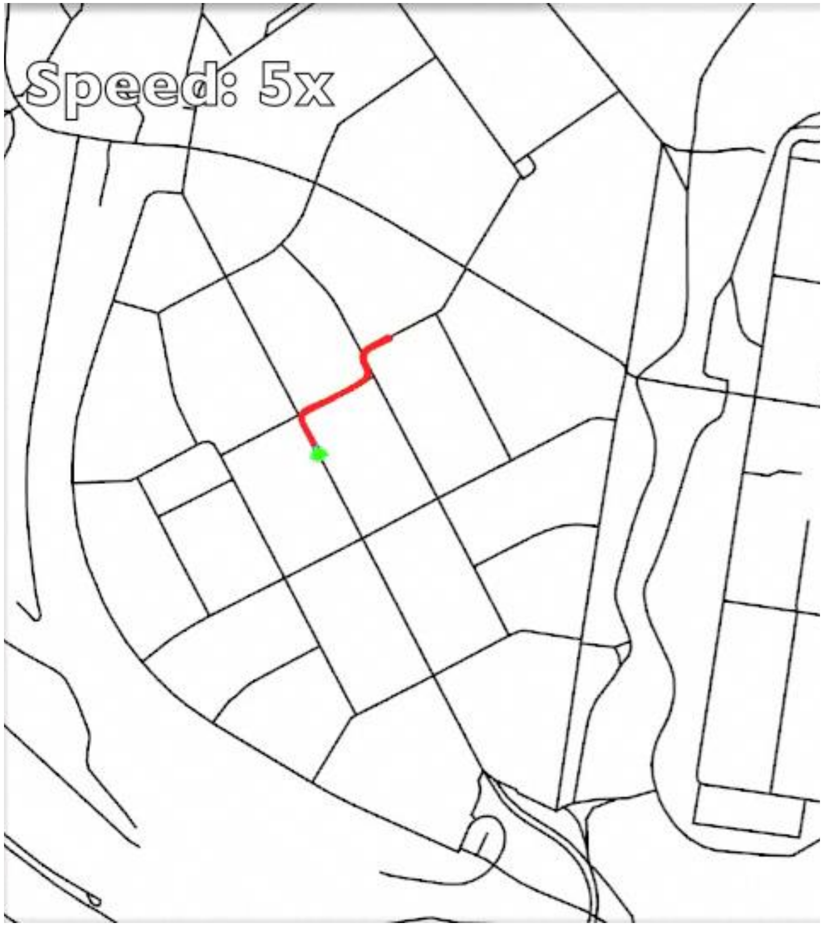
- Initialized from the global localization
- Works in a windowed fashion
- State: $\boldsymbol{x} = (x, y, \theta)$

Motion model:

- Visual Odometry

Observation model:

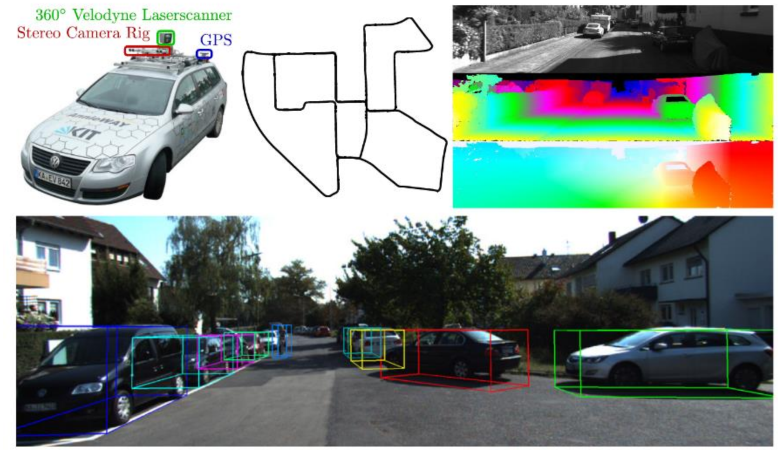
- Shape Matching
- $w^{[m]} = \lambda e^{-\lambda \cdot d_{CM}^{[m]}}$



Dataset

Dataset:

- KITTI suite benchmark
- Various urban and suburban sequences
- Accurate ground truth poses



For the experiments we used:

- 11 stereo sequences from KITTI with ground truth poses
- Manually downloaded OSM maps with ~1km radius each
- Visual Odometry (plain) used as baseline

Experimental Evaluation

Global localization:

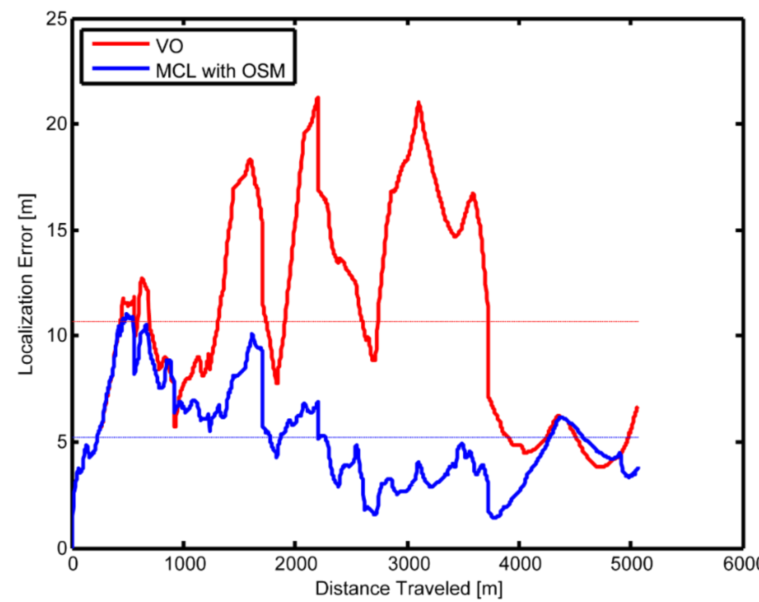
- Traveled distance, 400m
- Accumulated turning angle, π

Local path adjustment:

- Local window of 250 frames
- 500 particles

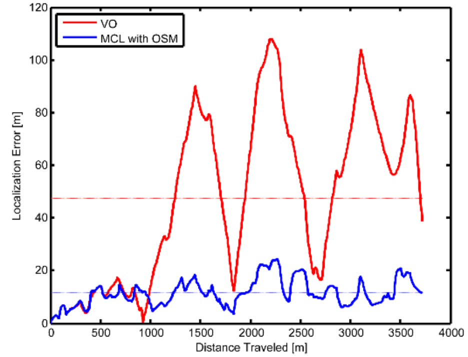
Computational time (average):

- 11.5s for map pre-processing (once)
- 15.6s for global localization (once)
- 0.02s for local path adjustment (every frame)

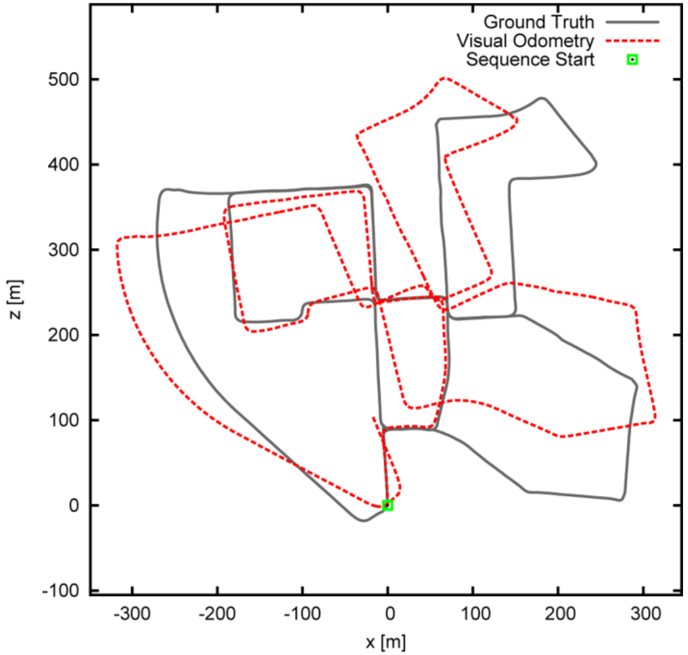


Results - Sequence 00

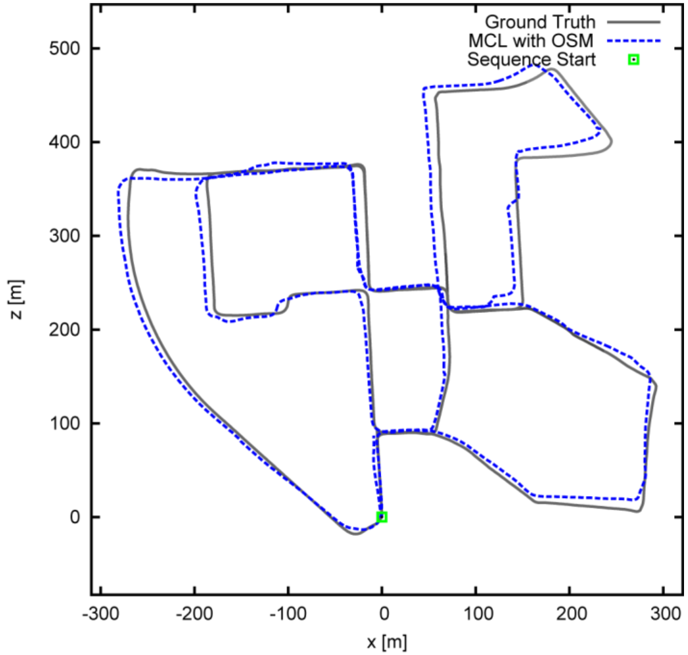
3.7 km traveled in a suburban area



Localization accuracy



Visual odometry

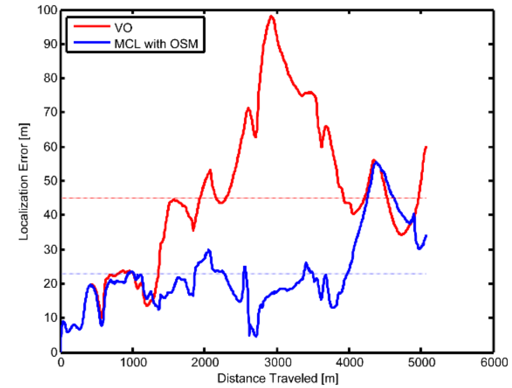


OpenStreetSLAM

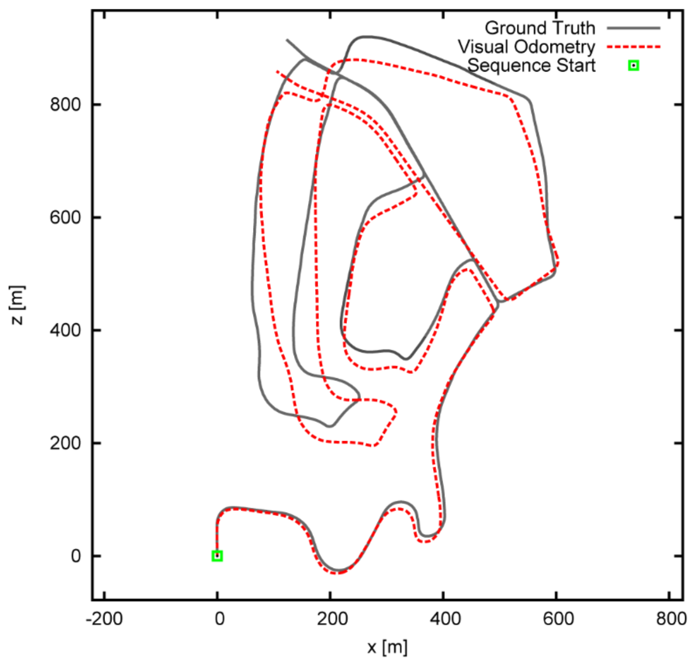
Computer Vision Group

Results - Sequence 02

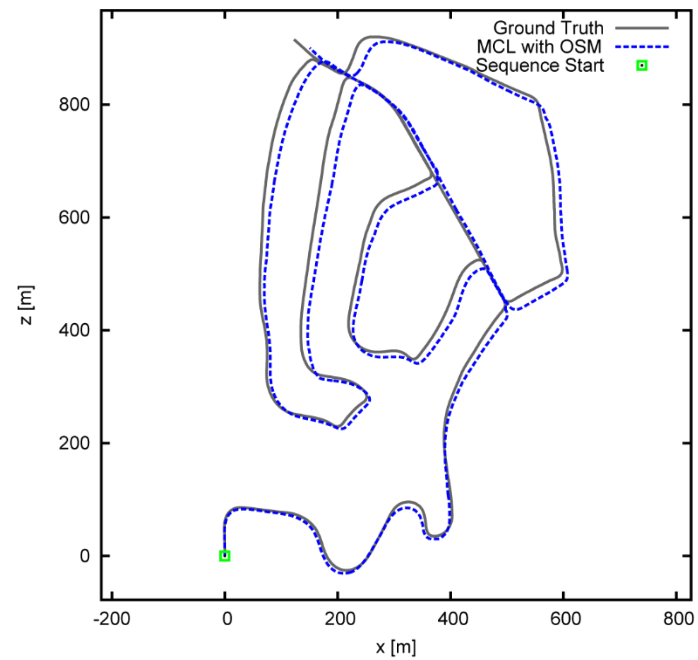
5.1 km traveled in a suburban area



Localization accuracy



Visual odometry

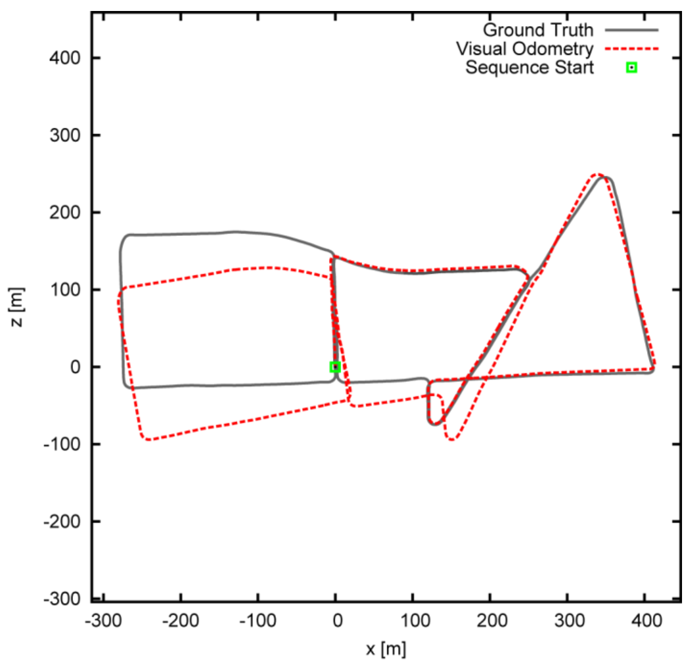


OpenStreetSLAM

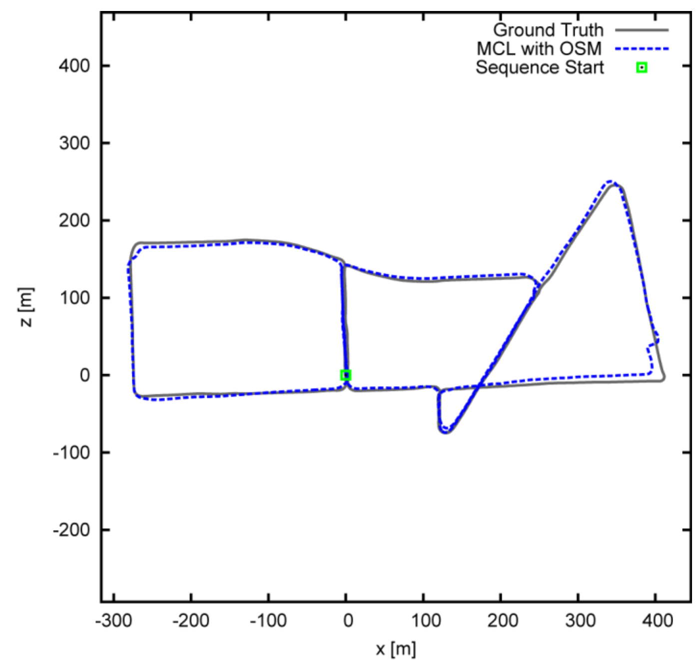
Computer Vision Group

Results – Sequence 13

Qualitative comparison to the state-of-the art

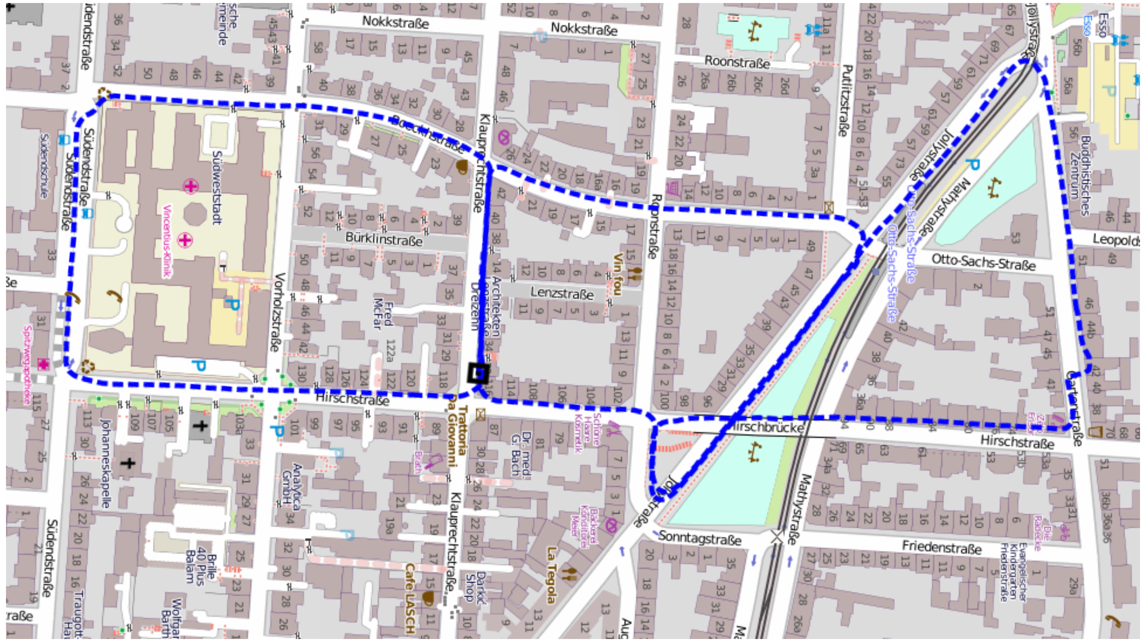


Beal et. al, IROS10



OpenStreetSLAM

Conclusion



- OSM maps for global localization and local trajectory adjustment
- Improved localization performance
- Generic, runs on top of any VO component
- Efficient, real-time capable
- No extra infrastructure needed

Computer Vision Group

RWTH AACHEN UNIVERSITY

Thank you!