# Online large-scale SLAM with stereo visual-inertial sensors

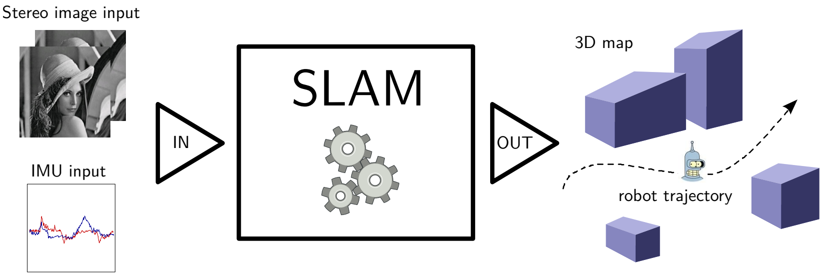
Dominik Schlegel
Supervisor: Prof. Dr. Giorgio Grisetti

November 9, 2015

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

SAPIENZA
UNIVERSITÀ DI ROMA

# Our mission




Required capabilities:

- Tracking (Landmark generation, Odometry)
- Local mapping (Key frame generation, Loop closing)
- Global mapping (Solving SLAM problem)

# Related work and state of the art

## Google (street) mapping

Car

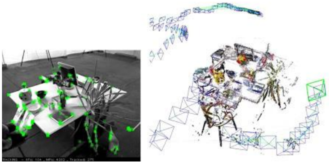
Trekker

Trolley

Snow mobile

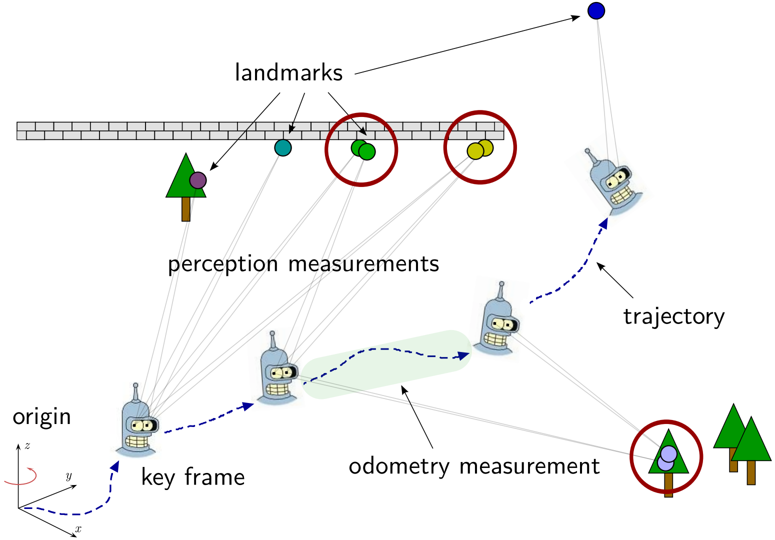
Trike

## ORB-SLAM (Mur-Artal et al)

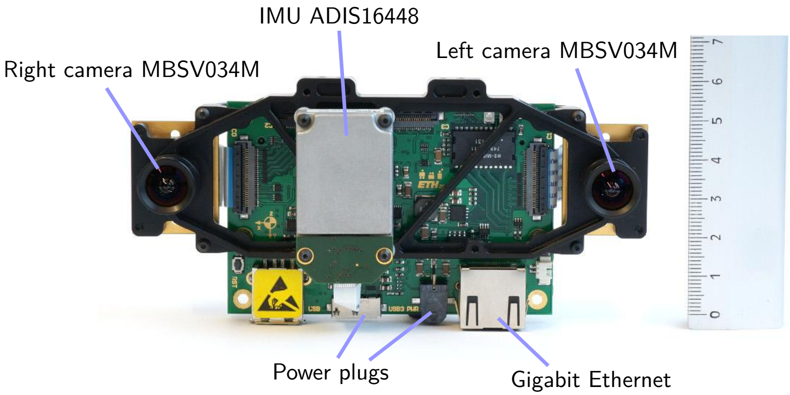


## ROVINA (Grisetti et al)

# Graph-based SLAM

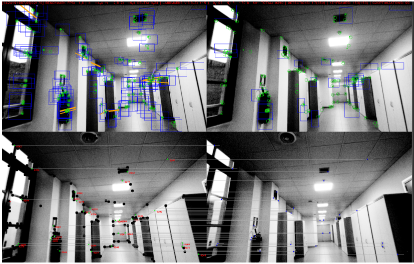


# Sensor setup: the VI-Sensor



# First steps

GUI: Visual tracking



GUI: Live map viewer

Many simplifications:

- Indoor datasets with slow steady movement
- Odometry by Alberto™
- Minimal range and disturbance

# Least squares state optimization

- StereoUV $\rightarrow$ Landmark position

  State: Positions $\mathbf{x}(x, y, z)$
  Measurements: Projections $p_L, p_R$

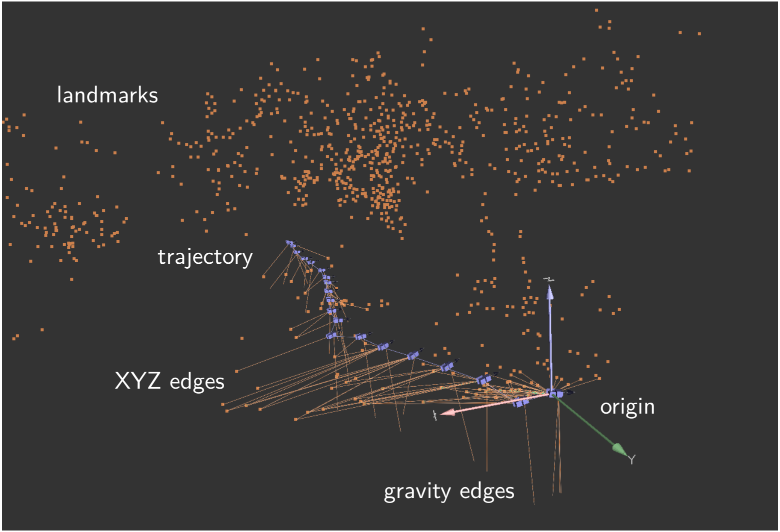
- StereoPosit $\rightarrow$ Odometry

  State: Transformation $T_{21}(R, \mathbf{t})$
  Measurements: Positions $\mathbf{x}$, projections $p_L, p_R$

- ICP (Besl et al) $\rightarrow$ Loop closures

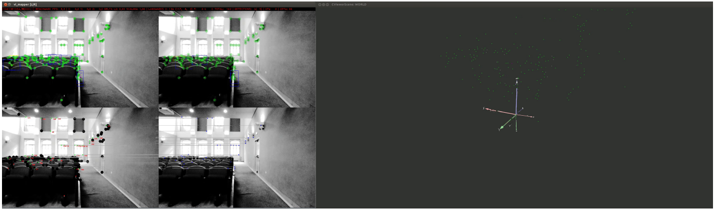
  State: Transformation $T_{BA}$
  Measurements: Positions $\mathbf{x}_A, \mathbf{x}_B$

# g2o: graph-based SLAM optimization (Kümmerle et al)

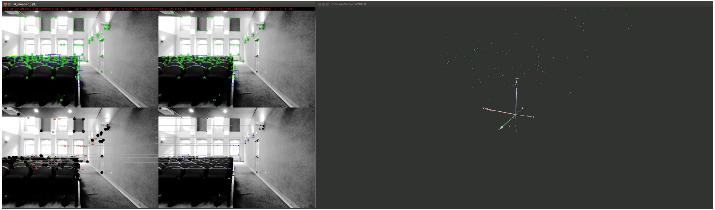


# System in action

Hand-held dataset: Aula Magna



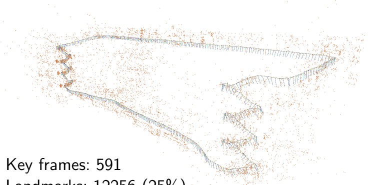
Bike mounted dataset: Streets in San Lorenzo



# Results: hand-held

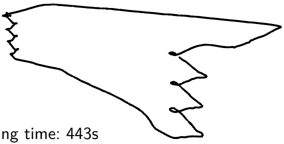
Stairs DIAG

Full graph



Key frames: 591
Landmarks: 12256 (25%)

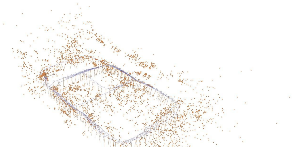
Trajectory



Recording time: 443s
Trajectory length: 394m

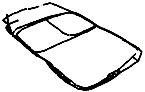
Aula Magna

Full graph



Key frames: 213
Landmarks: 4049 (22%)

Trajectory



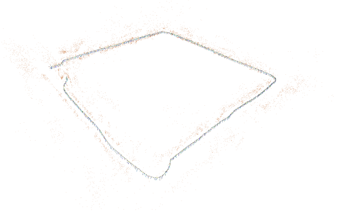
Recording time: 192s
Trajectory length: 126m

# Results: bike mounted

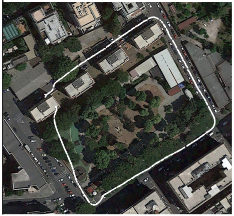
Full graph:

Trajectory (Google maps overlay):

Full graph:



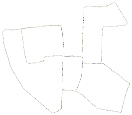
Trajectory (Google maps overlay):



# Results: KITTI

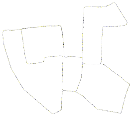
Full graph:

Trajectory:



Full graph:

Trajectory:



# Conclusions and final remarks

Regarding our final system:

- It works!
- It works well
- It works on a bike in streets and parks
- It works without IMU

What could've made things faster:

- Start off with mocked system
- Clear environment definition
- Deeper read-in phase

# Future work

Urgent:

- Filter integration (IMU)
- Parallelization
- Programmatic and cosmetic refactoring
- Open source publication

Optional:

- Improved feature handling
- Improved graph optimization