Msc in Artificial Intelligence and Robotics

Thesis Presentation

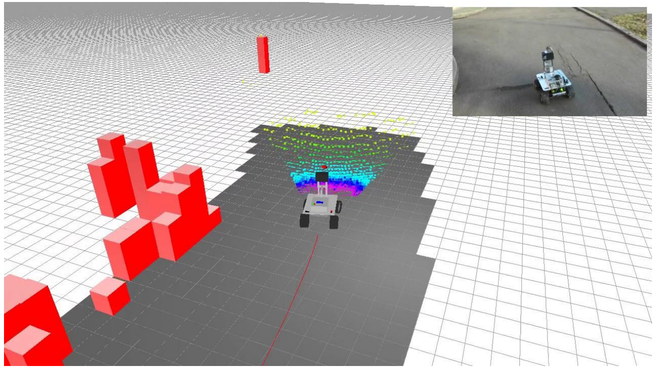
Sparse LiDAR Odometry using intensity channel: a comparison

A real-time front-end SLAM system for odometry estimation of a vehicle

Candidate	Francesco Starna
ID	1613660
Advisor	prof. Giorgio Grisetti

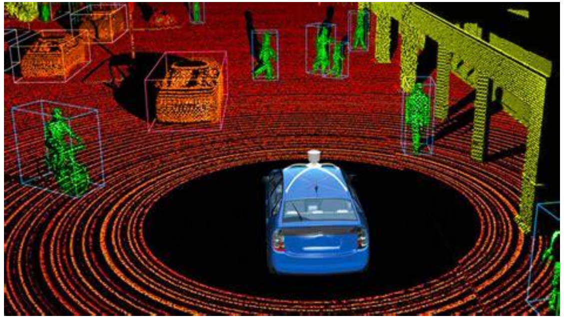
Sapienza University of Rome

Motivations



Robot navigation system

exploration, smart farming, surveillance, military, autonomous driving …

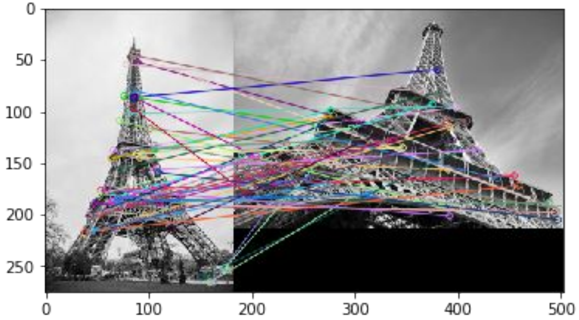


Modern sensors

laser scanners provide a dense representation of the environment

Variety of techniques

feature detection and matching, transform estimation

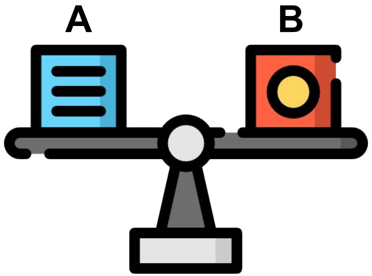


SAPIENZA

Project Goal

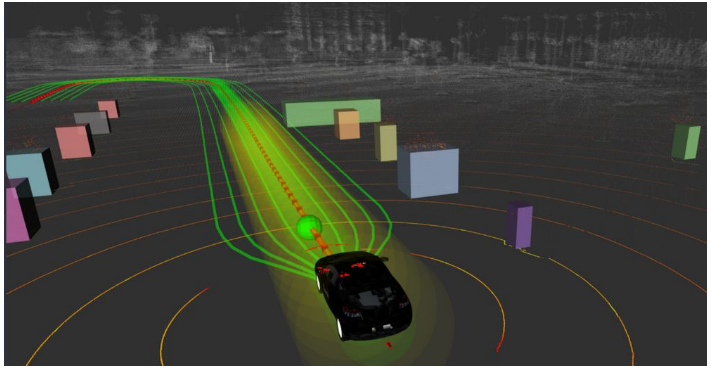
Methods comparison

highlighting differences between solutions available nowadays



Odometry estimation pipeline

building an entire tracking system which estimates the car trajectory along the path

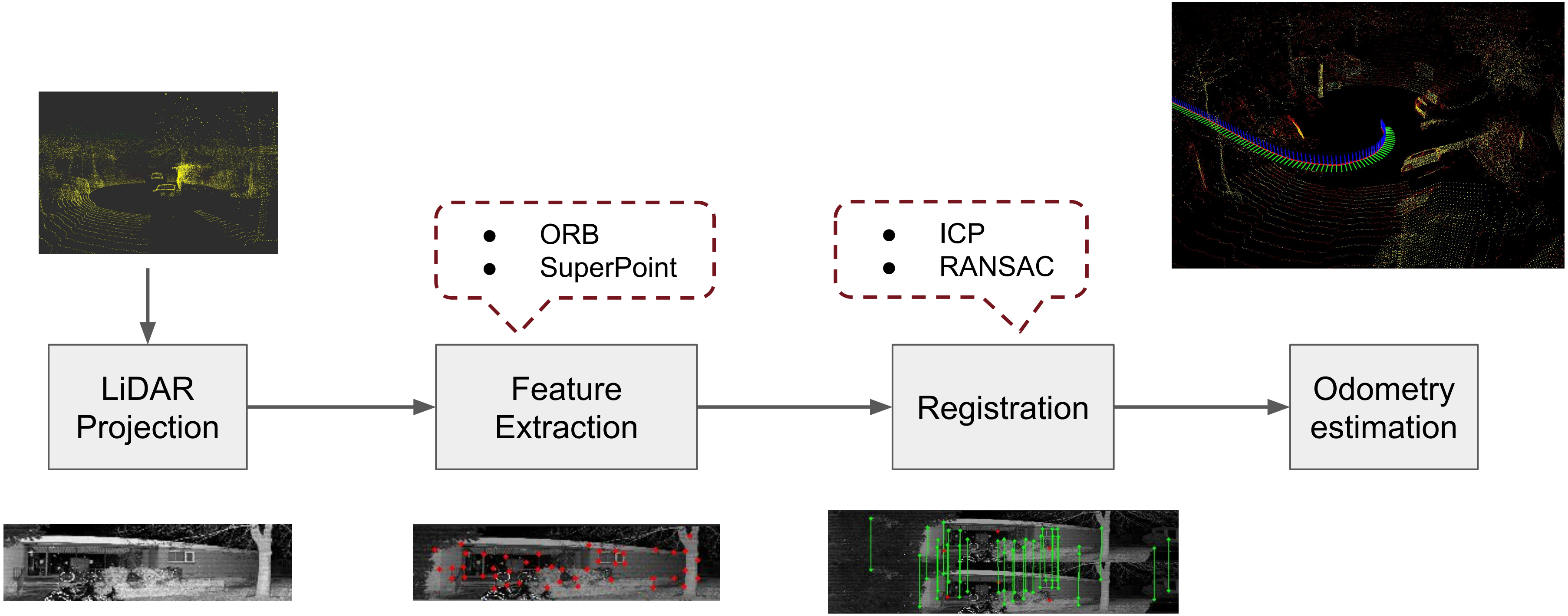


Real-time performance

entire process within the sensor frequency

SAPIENZA

Pipeline



ORB
SuperPoint
ICP
RANSAC
LiDAR Projection
Feature Extraction
Registration
Odometry estimation

LiDAR Projection

Projection | Extraction | Association | Registration | Odometry

IPB Car dataset

acquired in urban environment, with LiDAR and GNSS sensors

Projection function

from 3D Euclidean space to 2D image

$$r = \sqrt{x^2 + y^2 + z^2} \in \mathbb{R}$$

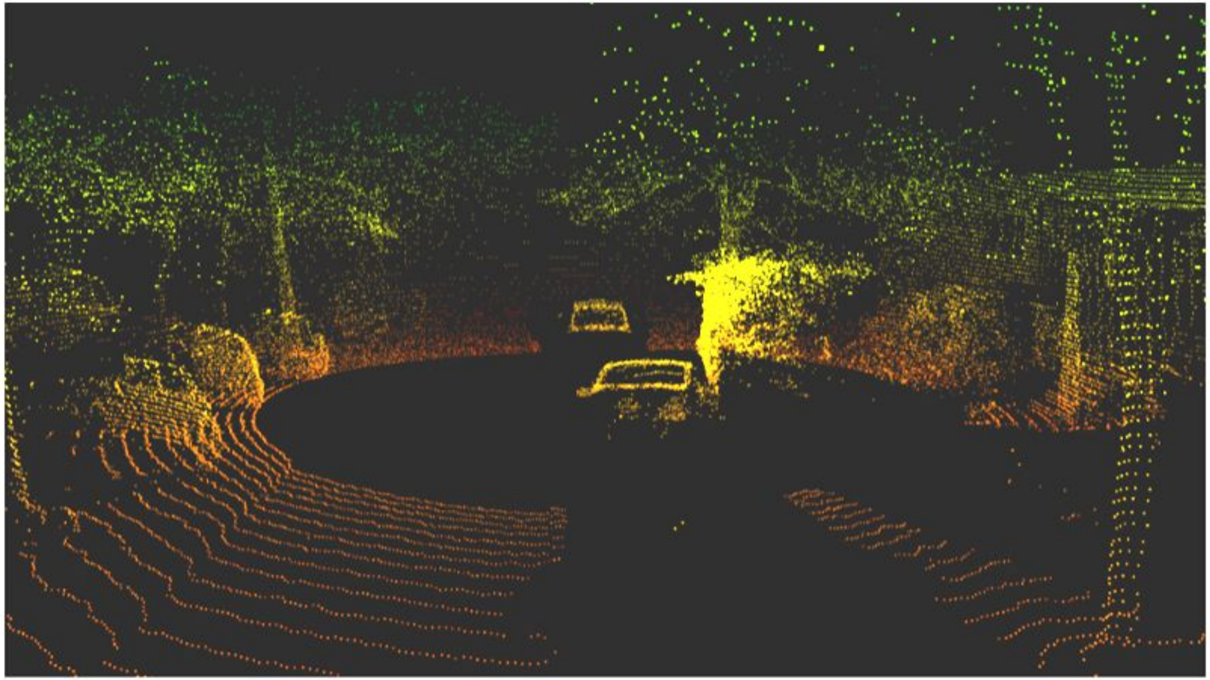
$$\theta = atan2(y, x) \in [-\pi/2, \pi/2]$$

$$\varphi = asin(z/r) \in [-\pi, \pi]$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{W}{2}(1 + \frac{\theta}{\pi}) \\ H \frac{f_{up} - \varphi}{f_{up} - f_{down}} \end{pmatrix}$$

Conversion issues

- multiple points may fall in the same pixel
- empty pixel



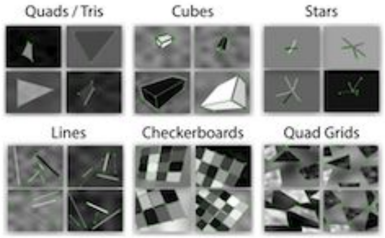
(a)

(b)

(c)

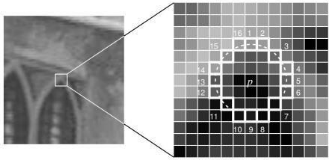
Feature Extraction

Projection | Extraction | Association | Registration | Odometry

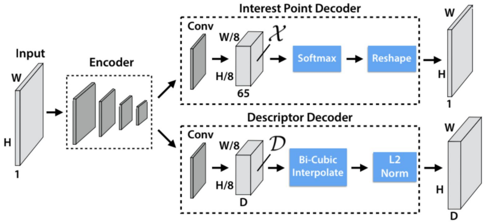


Traditional vs Deep Learning

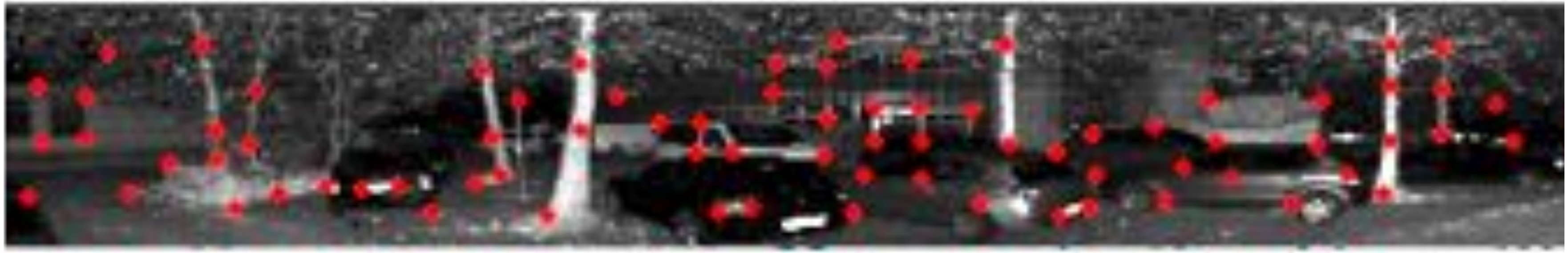
- Oriented FAST and Rotated BRIEF (ORB)
- SuperPoint

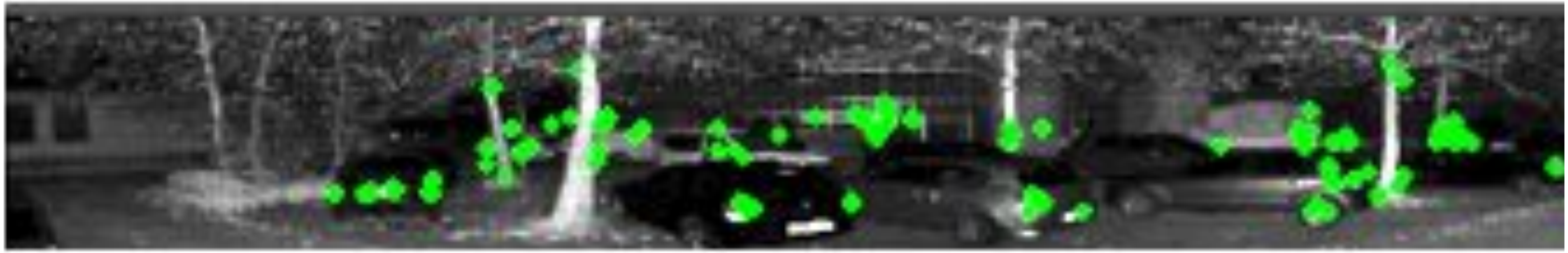


VS



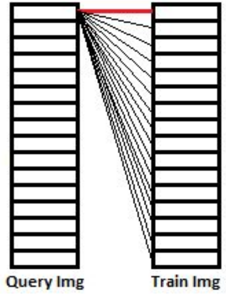
	Traditional	Deep Learning
Detection	Multiscale and oriented FAST detection of surrounding pixels	MagicPoint + Homographic Adaptation
Description	Gaussian kernel smoothing and binary test on image patch **128 bit vector**	semi-dense model + bi-cubic interpolation + L2 norm **256 float vector**





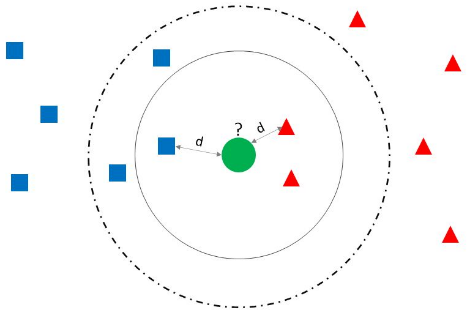
Data Association

Projection Extraction Association Registration Odometry



Brute-force approach

matching each descriptor with all the others



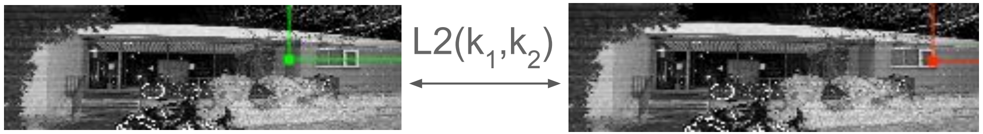
K-nearest neighbor (k=2)

choice is made by a ratio test

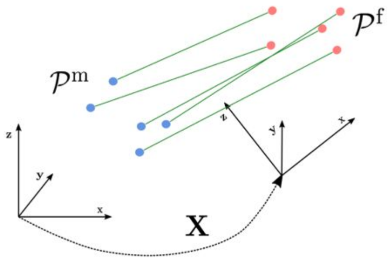
$$L(m_{k=1}, m_{k=2}) = \begin{cases} \text{keep} & \text{if } m_{k=1}.distance < m_{k=2}.distance * \text{ threshold)} \\ \text{discard} & \text{otherwise} \end{cases}$$

Euclidean norm check

looking for geometrical consistency



Transform Registration - Method



Projection — Extraction — Association — Registration — Odometry

Incremental approach

Iterative Closest Point (ICP)

- State $\quad \mathbf{X} = [\mathbf{R} \mid \mathbf{t}]$
- Perturbation $\quad \Delta\mathbf{x} = (\Delta x \; \Delta y \; \Delta z \; \Delta\alpha_x \; \Delta\alpha_y \; \Delta\alpha_z)$
- Operator $\quad \mathbf{X} \boxplus \Delta\mathbf{x} = v2t(\Delta\mathbf{x})\mathbf{X}$

- Measurement function $\quad \mathbf{h}_k^{icp}(\mathbf{X}) = \mathbf{R}^T(p_{j(k)}^m - \mathbf{t})$
- Error function $\quad \mathbf{e}_k^{icp}(\mathbf{X}) = \mathbf{h}_k^{icp}(\mathbf{X}) - p_{j(k)}^f$
- Jacobian $\quad \mathbf{J}_k^{icp}(\mathbf{X}) = \left(\mathbf{I} \; - [p_{j(k)}^m]_\times\right)$

Direct approach

Closed-form solution to Least-Square (SVD)

Algorithm 2 Closed-form Solution of Least-Squares Optimization

Require: point correspondences $\{\langle x_0, y_0\rangle, ..., \langle x_K, y_K\rangle\}$

Ensure: closed-form solution of $\mathbf{R}$ and $\mathbf{t}$

$\boldsymbol{\mu}_x \leftarrow \frac{1}{k}\sum_{i=1}^k \mathbf{x}_i, \quad \boldsymbol{\mu}_y \leftarrow \frac{1}{k}\sum_{i=1}^k \mathbf{y}_i$

$\boldsymbol{\sigma}_x \leftarrow \frac{1}{k}\sum_{i=1}^k ||\mathbf{x}_i - \boldsymbol{\mu}_x||^2, \quad \boldsymbol{\sigma}_y \leftarrow \frac{1}{k}\sum_{i=1}^k ||\mathbf{y}_i - \boldsymbol{\mu}_y||^2$

$\boldsymbol{\Sigma} \leftarrow \frac{1}{k}\sum_{i=1}^k (\mathbf{y}_i - \boldsymbol{\mu}_y)(\mathbf{x}_i - \boldsymbol{\mu}_x)^T$

$\mathbf{UDV}^T \leftarrow SVD(\boldsymbol{\Sigma})$

if $det(\mathbf{U})det(\mathbf{V}) = 1$ **then**

  $\mathbf{S} = \mathbf{I}_{3\times 3}$

else

  $\mathbf{S} = diag(1, 1 - 1)$

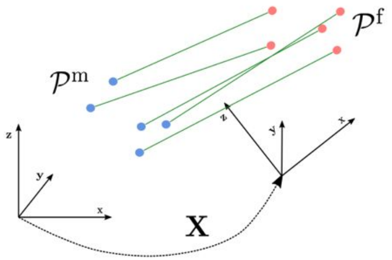
end if

$\mathbf{R} \leftarrow \mathbf{USV}^T$

$\mathbf{t} \leftarrow \boldsymbol{\mu}_y - \mathbf{R}\boldsymbol{\mu}_x$

return $\mathbf{R}, \mathbf{t}$

Transform Registration - Outlier Rejection



Projection — Extraction — Association — Registration — Odometry

Incremental approach

Robust estimator

- L1 Omega norm of the error function
- New error term
- Reducing contributions of higher errors

$$\underset{X}{argmin} \sum_{k=1}^{m} \rho(u_k(x))$$

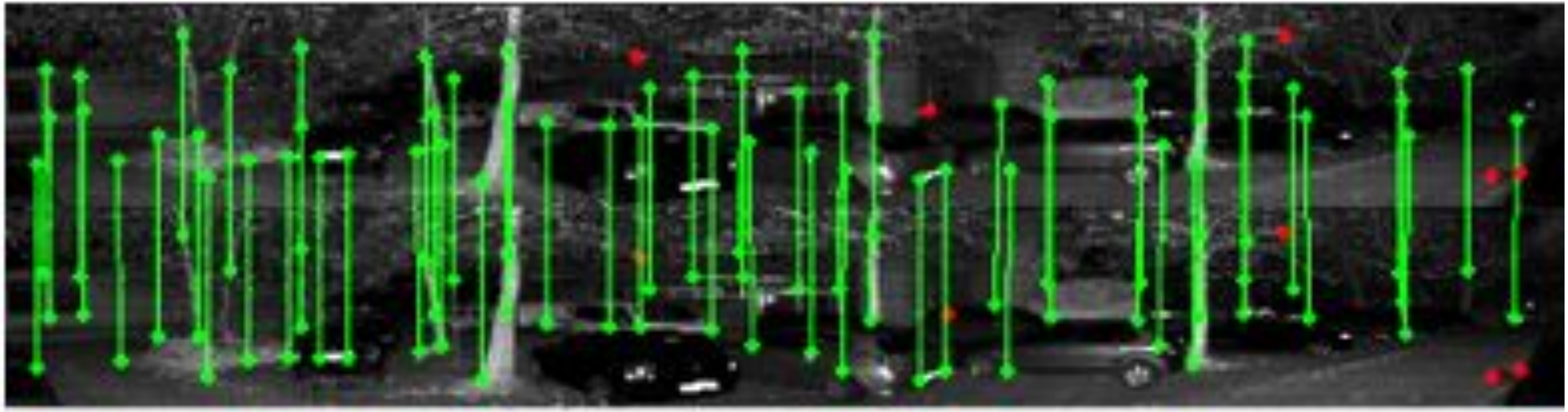
$$\begin{aligned}
&\mathbf{e}_k \leftarrow \mathbf{h}_k(\check{\mathbf{x}}) \boxminus \mathbf{z}_k \\
&\mathbf{J}_k \leftarrow \frac{\partial \mathbf{e}(\check{\mathbf{x}} \boxplus \Delta\mathbf{x})}{\partial \Delta\mathbf{x}}\Big|_{\Delta x=0} \\
&u_k \leftarrow \sqrt{\mathbf{e}_k(\mathbf{x})^T \Omega_k \mathbf{e}_k(\mathbf{x})} \\
&\gamma_k \leftarrow \frac{1}{u_k}\frac{\partial \rho_k(u)}{\partial u}\Big|_{u=u_k} \\
&\Omega_k = \gamma_k \Omega_k \\
&\mathbf{H} \leftarrow \mathbf{H} + \mathbf{J}_k \Omega_k \mathbf{J}_k \\
&\mathbf{b} \leftarrow \mathbf{b} + \mathbf{J}_k \Omega_k \mathbf{e}_k
\end{aligned}$$

Direct approach

Random Sample Consensus (RANSAC)

1. Randomly select a minimum amount of data
2. Compute a model (closed-form)
3. Evaluate the model (Euclidean distance)
4. Repeat the procedure k times

iterations: $$k = \frac{log(1-p)}{log(1-w^n)}$$



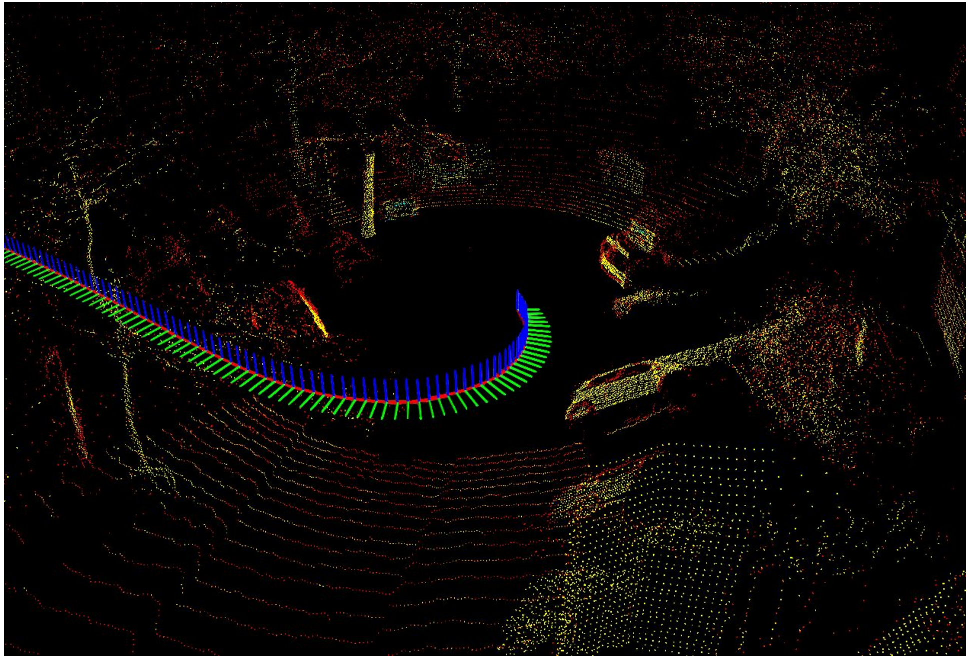
Odometry Estimation

Projection — Extraction — Association — Registration — Odometry

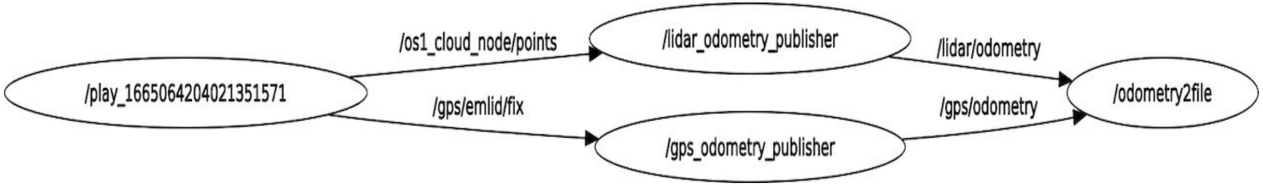
Accumulation of transforms

frame by frame matrix multiplication

$$\mathbf{T}_k^0 = \mathbf{I}_{4\times4}\ \mathbf{T}_1^0\ \mathbf{T}_2^1\ \ldots\ \mathbf{T}_k^{k-1}$$

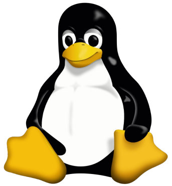


Experiments



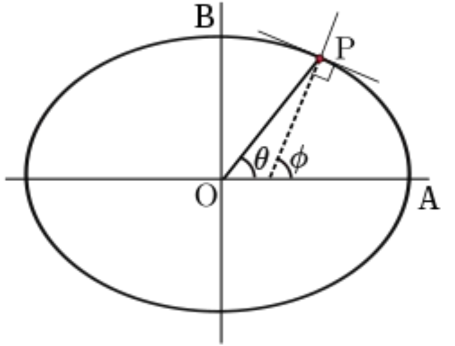
Hardware

- Ubuntu 20.04, C++
- 8-core Intel core i9 without graphic card
- LiDAR OS1-64 vertical, 1024 horizontal, 10Hz, 65536 points per frame



GNSS estimation

conversion from geodetic to geocentric coordinates



$$x = R_{Heart}(\mathbf{P}) \cdot cos(long) \cdot cos(\mathbf{gLat}(lat))$$
$$y = R_{Heart}(\mathbf{P}) \cdot sin(long) \cdot cos(\mathbf{gLat}(lat))$$
$$z = R_{Heart}(\mathbf{P}) \cdot sin(\mathbf{gLat}(lat))$$

Robot Operating System

ROS middleware for playing dataset and run pipeline

Evaluation

Root Mean Square Error

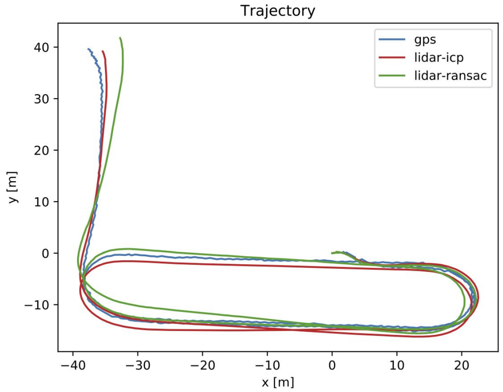
$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k} ||\hat{\mathbf{t}}_k - \mathbf{t}_k^{GNSS}||^2}$$

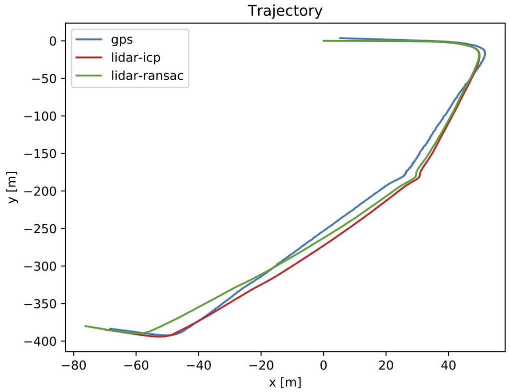
SAPIENZA

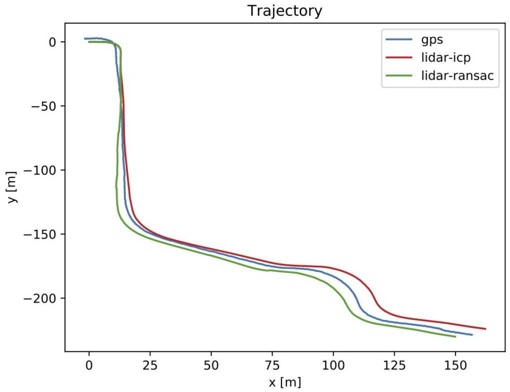
Parameters Tuning

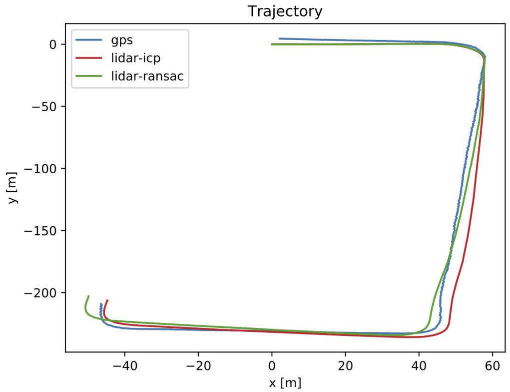
Feature Extraction			Tracking		
SuperPoint	nFeatures	300	BF Matcher	knnThreshold	0.7
	threshold	0.1		normType	L2/H
	nmsBlockSize	6		normThreshold	30
FAST	threshold	30	**Registration**		
ORB	nFeatures	300	IRLS	iterations	1
	scaleFactor	1.1		kernelThreshold	5e-5
	nLevels	8		damping	0.5
	edgeThreshold	15	RANSAC	iterations	30
	patchSize	15		inliersThreshold	20cm

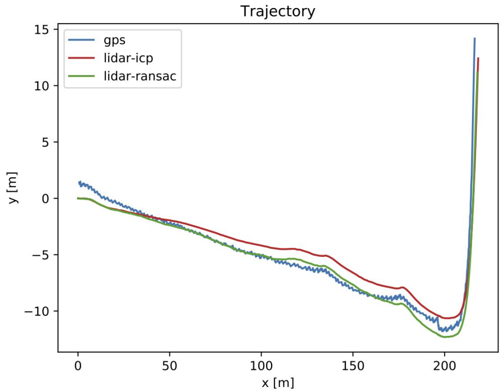
Results

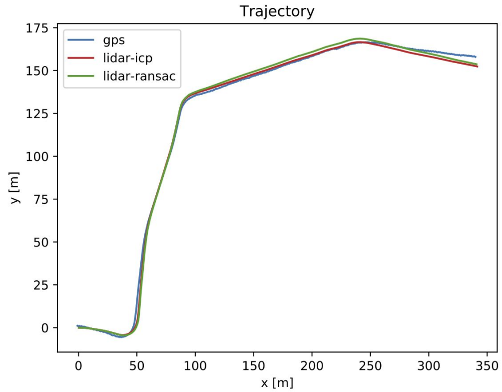

Trajectory
gps
lidar-icp
lidar-ransac
x [m]
y [m]


Trajectory
gps
lidar-icp
lidar-ransac
x [m]
y [m]


Trajectory
gps
lidar-icp
lidar-ransac
x [m]
y [m]


Trajectory
gps
lidar-icp
lidar-ransac
x [m]
y [m]


Trajectory
gps
lidar-icp
lidar-ransac
x [m]
y [m]


Trajectory
gps
lidar-icp
lidar-ransac
x [m]
y [m]

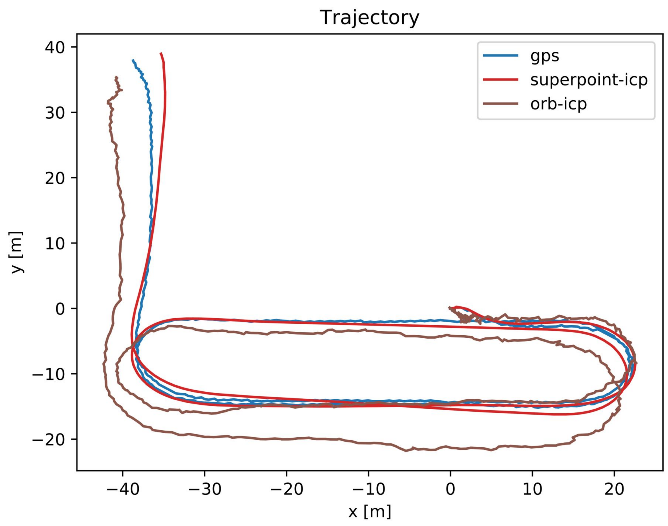
Comparison

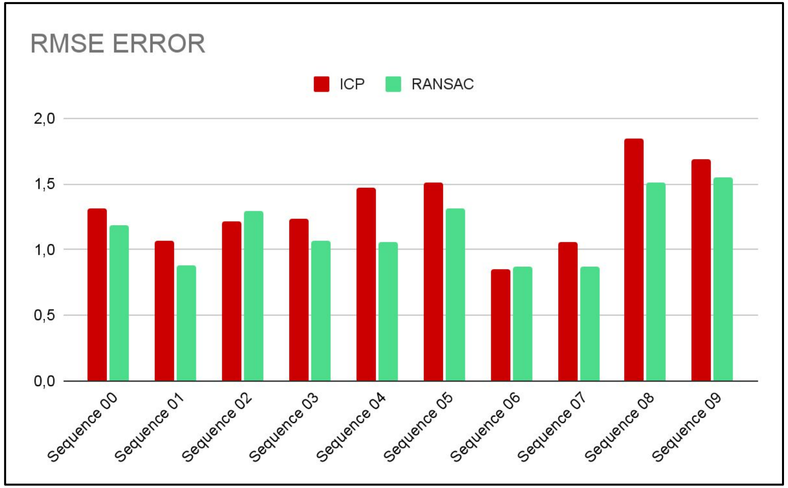
SuperPoint > ORB
(with ICP)

- SuperPoint detections sparse and repeatable
- ORB detections poorly distributed
- ORB 110HZ >> SuperPoint 20Hz

RANSAC > ICP
(with SuperPoint)

- RANSAC more accurate
- ICP 70 kHz >> RANSAC 20kHz
- ICP no longer valid in global optimization
- RANSAC suitable for loop closure





Conclusions

- ➢ **Feature extraction**

  Machine learning is overcoming traditional approaches

- ➢ **Transform estimation**

  probabilistic framework more accurate than least-square optimization and more suitable for other task in SLAM frameworks

- ➢ **Hardware requirements**

  machine learning technique feasible also without graphic card

Future Work

- Use of LiDAR depth information in conjunction with intensity
- Registration in non-successive frame
- Loop closure detection and bundle adjustment

Thanks