Matching Heterogeneous Sensing Pipelines to Digital Maps For Ego-Vehicle Localization

PhD Dissertation by Augusto Luis Ballardini

Supervisor: Prof. Domenico G. Sorrenti

Tutor: Prof. Enza Messina

Università degli Studi Milano – Bicocca

PhD XXVIII Cycle 27th March 2017

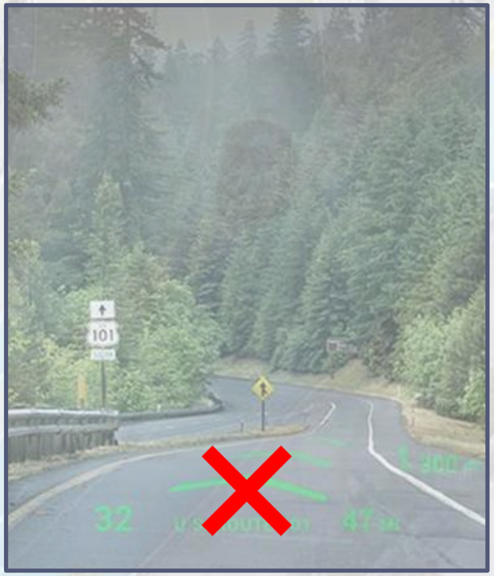
Presentation Outline

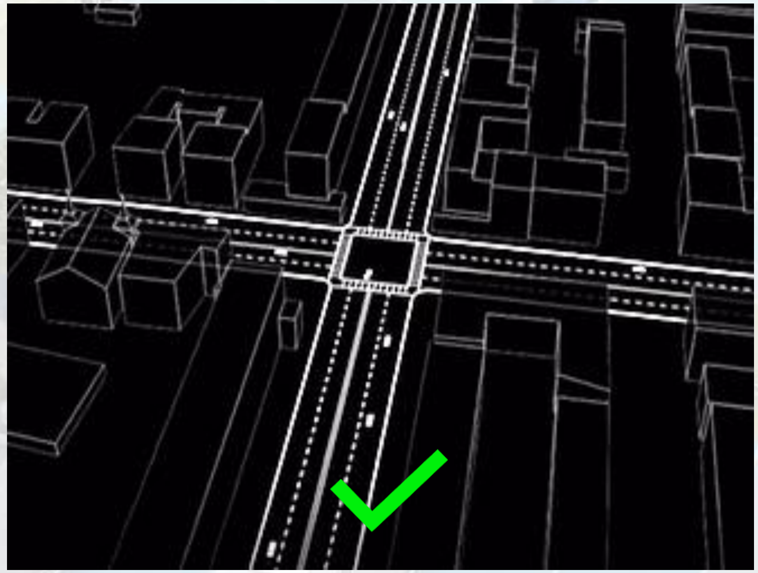
Matching Heterogeneous Sensing Pipelines to Digital Maps for Ego-Vehicle Localization

- *Aim of the research*
- *Localization with standard robotics techniques*
- *Overview of the Proposed Approach*
- *Discussion, Assessment and Results*
- *Conclusions*

Aim of my Research

Research for novel techniques to allow an autonomous vehicle to self-drive, safely and reliably, in an urban scenario





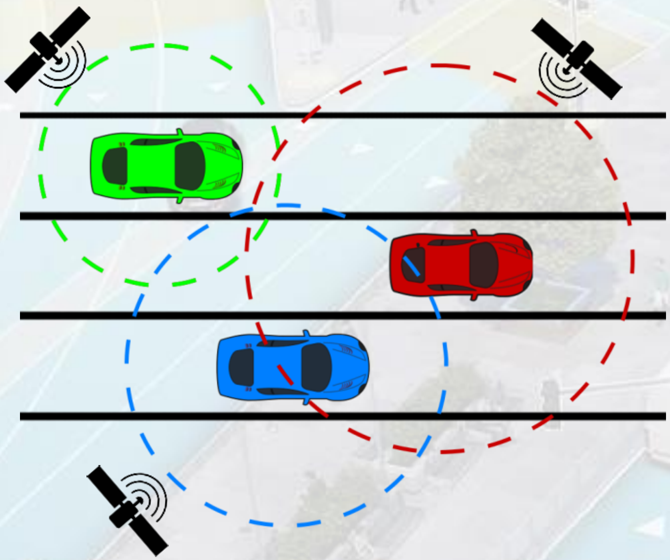
Strategy
Path / Trajectory
Planning

How to understand nearby scene in order to LOCALIZE a vehicle

Driving Assistance
ADAS V2X

Introduction 1/3 - Localization

Localization plays a key role for autonomous systems



Introduction 2/3 - Localization

- GPS systems in urban scenarios have **availability** and **reliability** limitations
- State of the art, used solutions so far
  - *ad-hoc* and *pre-built* maps
  - place recognition methods
- Leverage information from mapping services

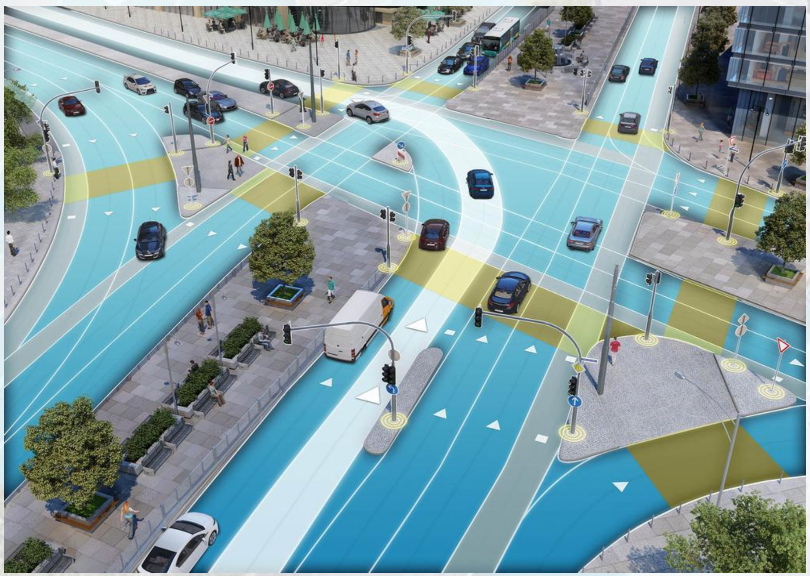
Credits: Niko Sünderhauf

Credits: 360.here.com

Introduction 3/3 - Localization

“Humans are able to use a map, combined with visual input and exploration, to localize effectively”

LOST! Leveraging the Crowd for Probabilistic Visual Self-Localization - Brubaker, Geiger, Urtasun 2013



Credits: 360.here.com

Credits: 360.here.com

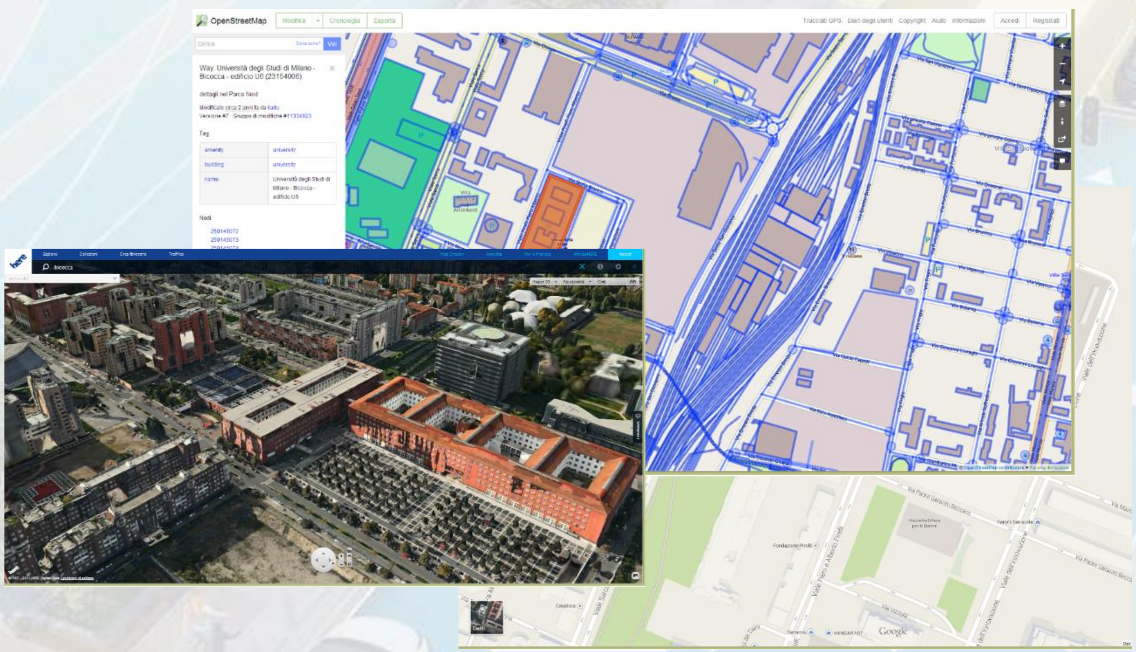
Proposed a

<u>PROBABILISTIC FRAMEWORK</u>

for *“Road Layout Estimation”* that leverages <u>Existing Maps</u>

Using Existing Maps: advantages

- **Maps** - continuously updated by the community
- **Strong prior** - no need to map the environment
- Validate/update/integrate the maps with the data provided by vehicle sensors



- *Localize*
- *Map*
- *Collaborative*

Road Layout Estimation (RLE)
Part 1 - Layout Hypotheses

Layout Hypothesis – LH represents a **description of the vehicle state** and an estimate of the surrounding scene

Exploits information generated by any kind and any number of sensors, as well as the information from the maps

Credits: 360.here.com

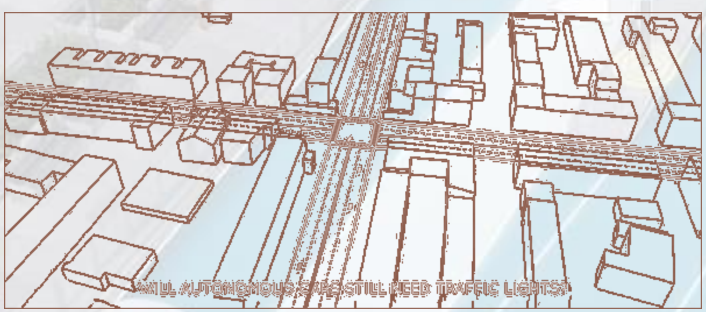
Credits: 360.here.com

Road Layout Estimation (RLE) Part 2 - Layout Components

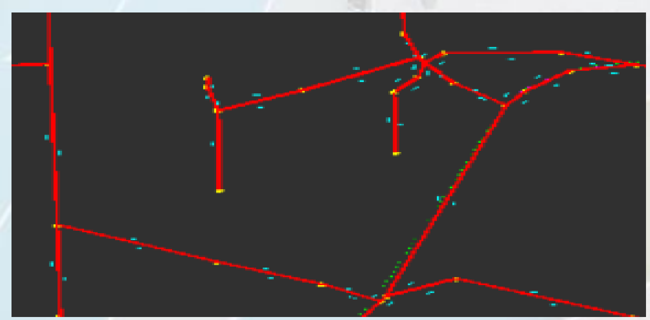
- ***Layout Components – LC*** are associated to ***Layout Hypotheses***

- Layout Components describe elements of the surrounding scene

- We initialize the LC using the output of external modules or detectors
  The detections can be physical, e.g. a road marking, a building etc., or virtual e.g. a measure of the current number of lanes in the street



Building Outlines / Credits: 360.here.com



OpenStreetMap Road Graph

Road Layout Estimation (RLE)
Layout Manager

Layout Hypothesis and Layout Components are handled with a particle filtering approach

Prediction Step

- Predict new LHs state, using velocities
- Update LC according to the new state

Importance Weights

- Score the LHs

Resampling

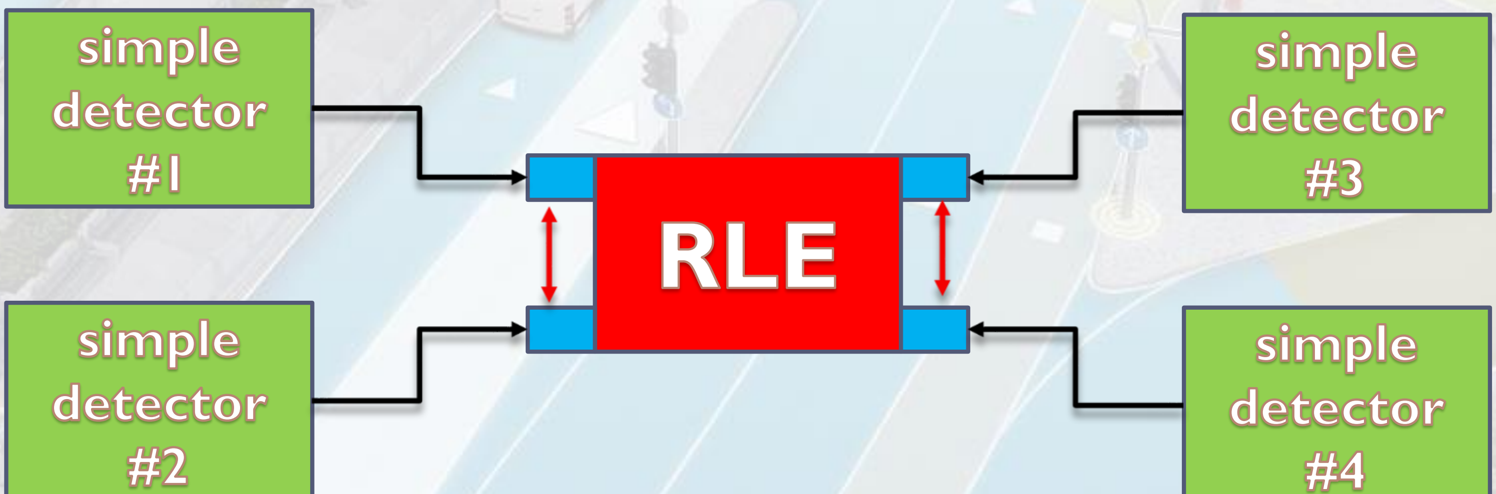
- Create a new set of LH using the most likely ones

Road Layout Estimation

Main Architectural Insights

- differences with respect to the state of the art approaches -

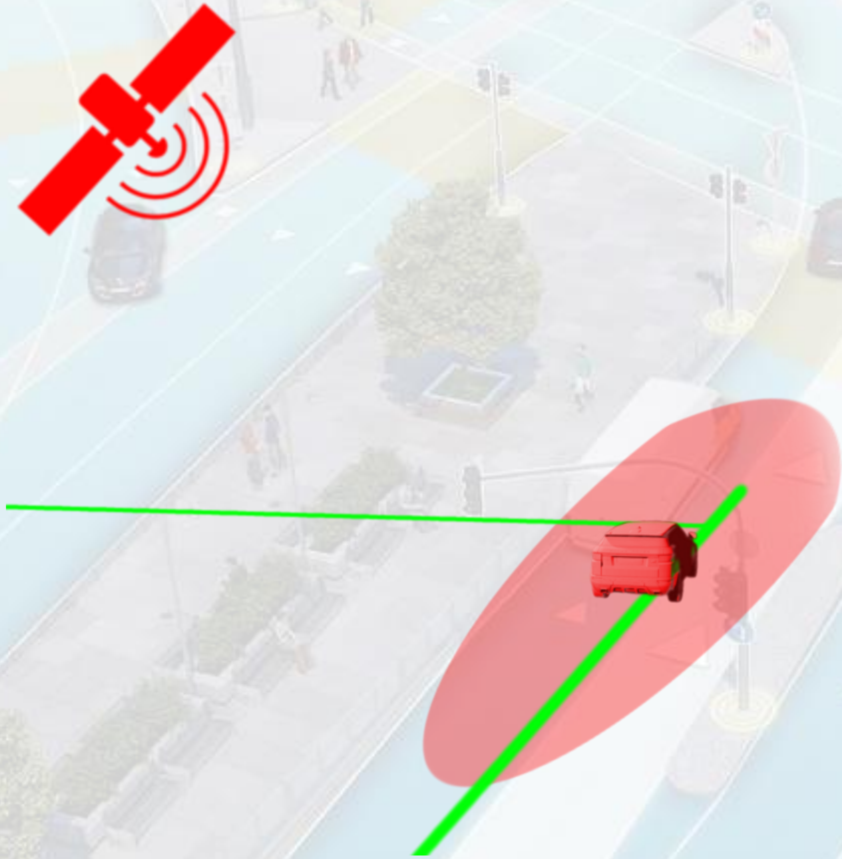
- **Seamless Integration** *of external detectors by means of Layout Components wrappers*
- **Easiness** of changing The Inference Structure: no sensor-set defined a-priori
- **Interaction schemes** *Layout Components allow us to cope with complex scenarios*
- **Decoupling Frequency** between Framework and Detectors for better performances



Layout Components

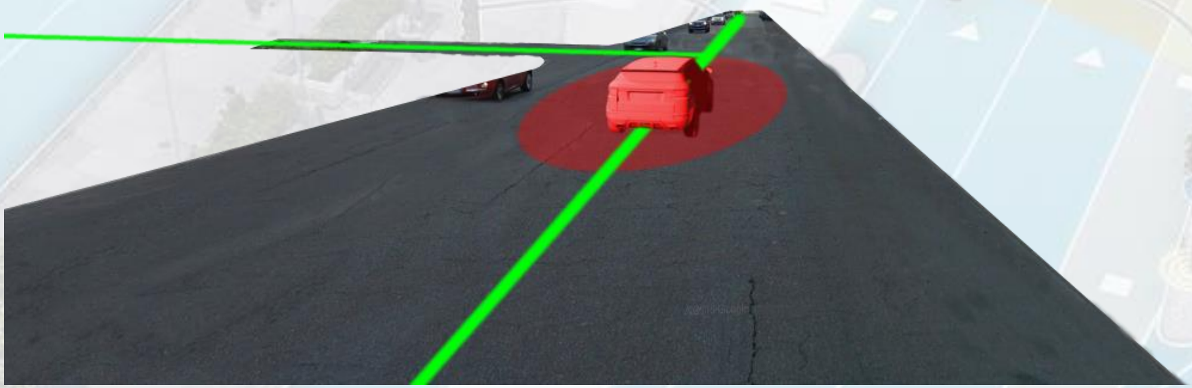
A GPS signal only gives position information with uncertainties of the order of 10-20 meters

Layout Components



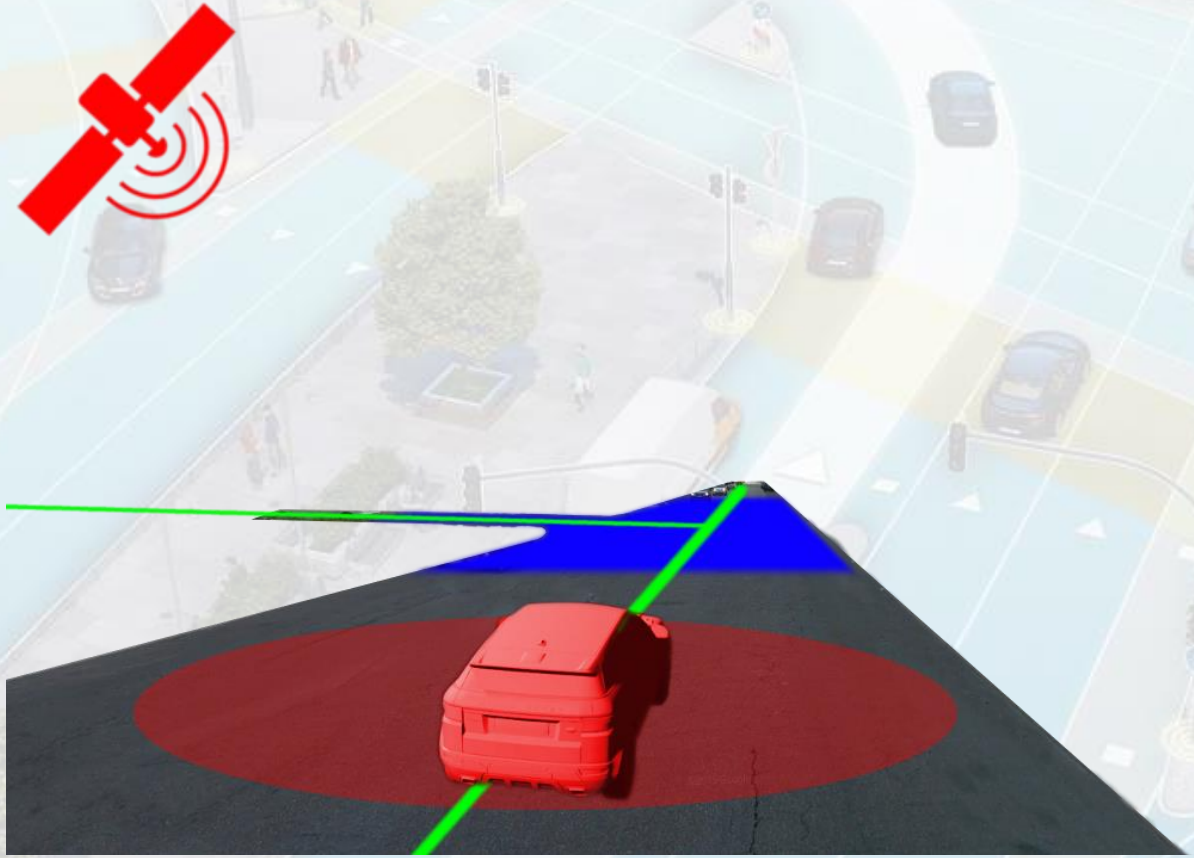
Including a road graph in the localization process can help the system to reduce the uncertainty achiving in-road level accuracies

Layout Components



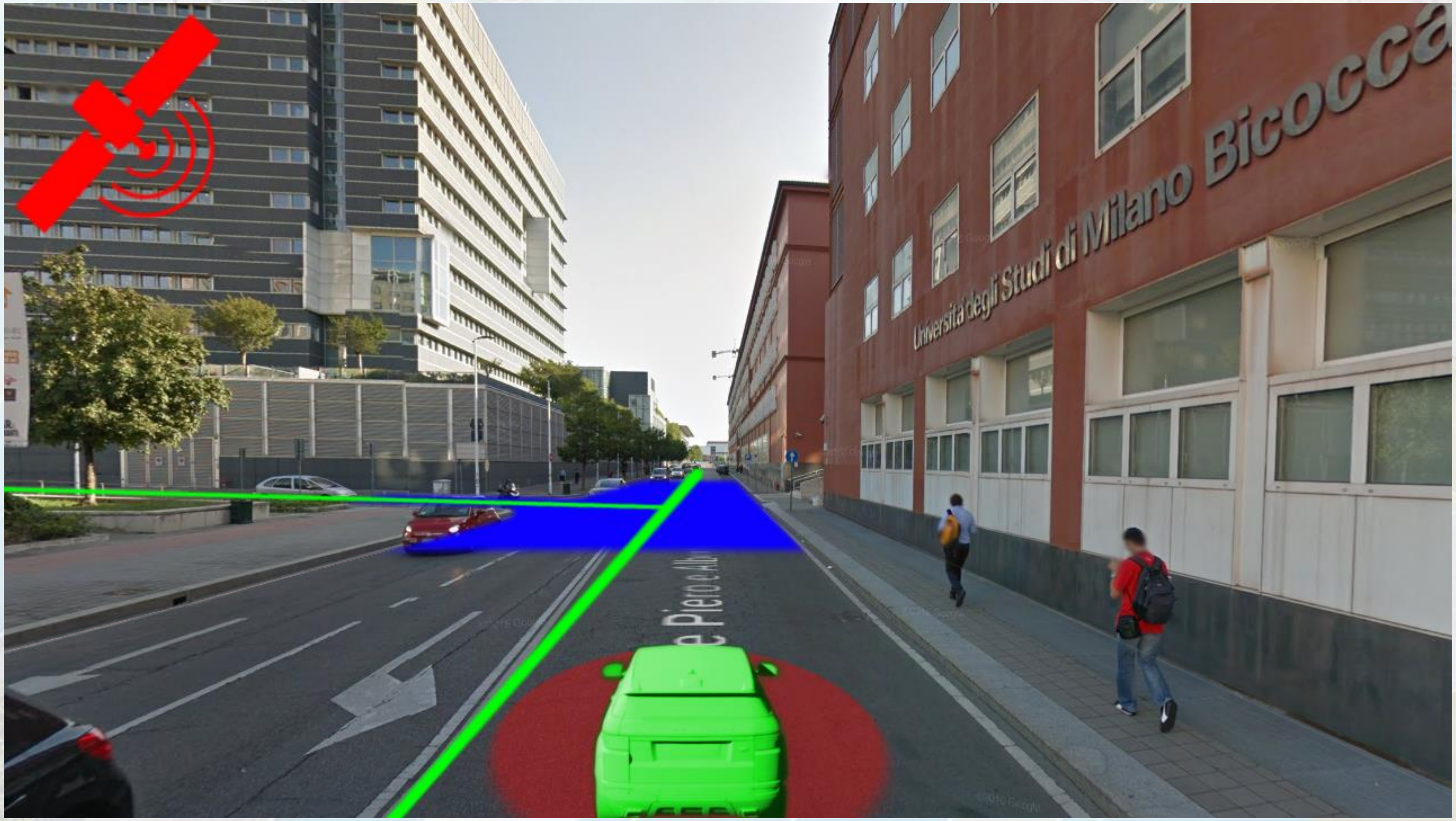
If the system is able to detect the road surface and its boundaries

Layout Components



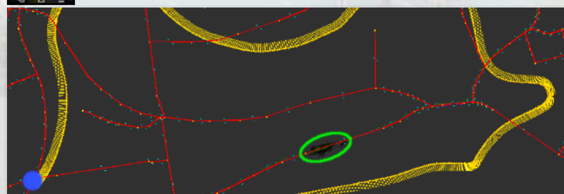
If the system is able to detect the road surface and its boundaries we can detect intersection areas, reducing the longitudinal uncertainties

Layout Components



Adding building's façade detection would result in a lateral localization improvement

Proposed Layout Components



OpenStreetMap **component**

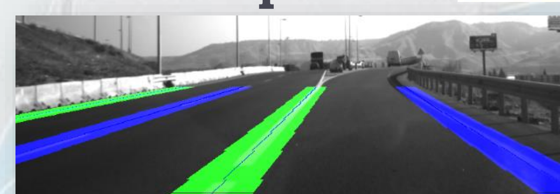
Road Intersection **component**

ROAD LAYOUT ESTIMATION

RLE

Buildings **component**

Road Lane **component**

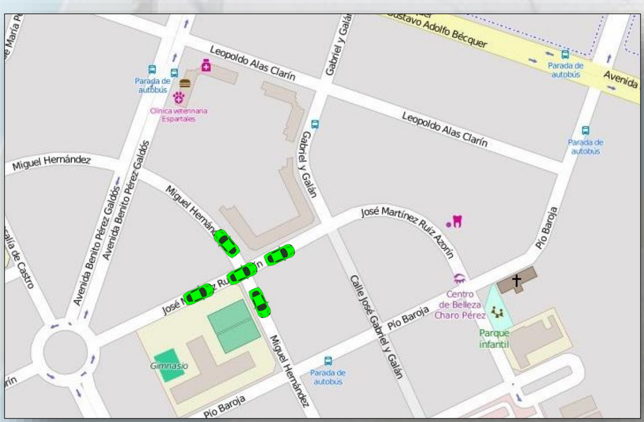


Road Width **component**

OpenStreetMap Component

- Having the following sensors…
  - GPS (*)
  - Stereo-rig and Visual Odometry module (LIBVISO2) for vehicle speed
  - OpenStreetMap module (that handles the road segments)

- Layout Hypotheses “ ” state is composed of:
  - Vector of LCs = { ***OpenStreetMap + Vehicle Position/Speed*** }
  - Vehicle Motion Model
  - Score of the hypothesis



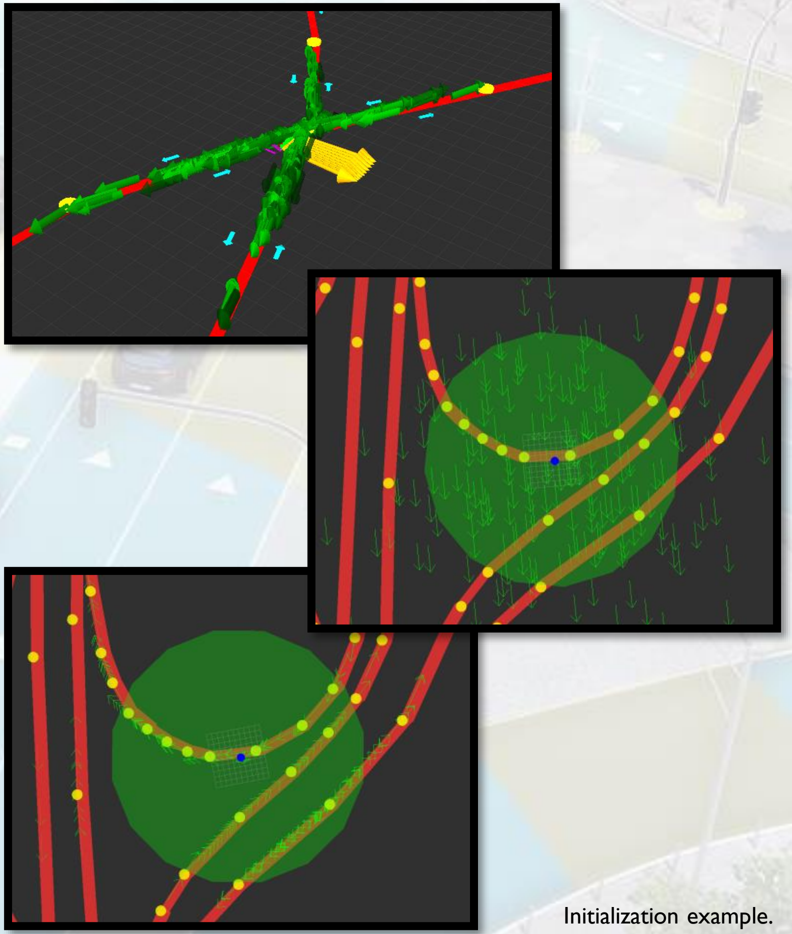
* GPS is used for ***initialization only***, i.e., to narrow the starting localization uncertainty into a “*feasible*” area

OpenStreetMap Component
1/3 Initialization Phase

The framework uses a GPS fix to download a map from OSM and to initialize the hypotheses poses (position+orientation)

A *"lock-on-road"* procedure to the nearest road segment is performed

Without a GPS fix at start time, a global localization task is executed (in a pre-cached area/zone)

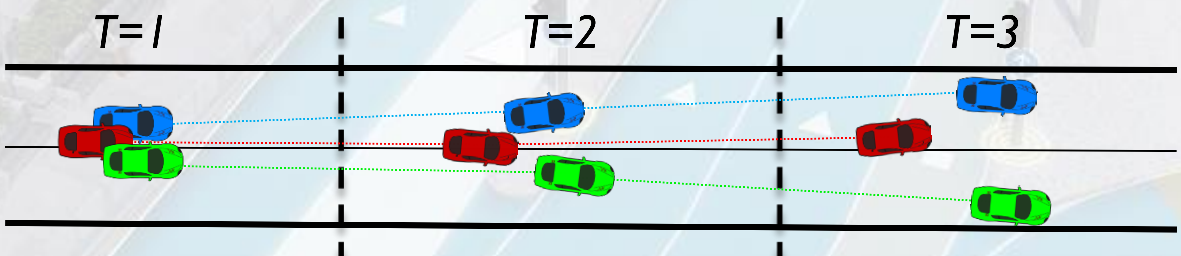


Initialization example.
Red segments = roads
Green arrows = pose hyp

OpenStreetMap Component
2/3 Prediction Step

For each hypothesis, the Layout Manager **updates**

1. The Position of the LH, using a simple motion model *(velocities plus uncertainties)*
2. Velocities of the LH: same velocity plus error
3. Call Layout Components update routines *(OpenStreetMap)*

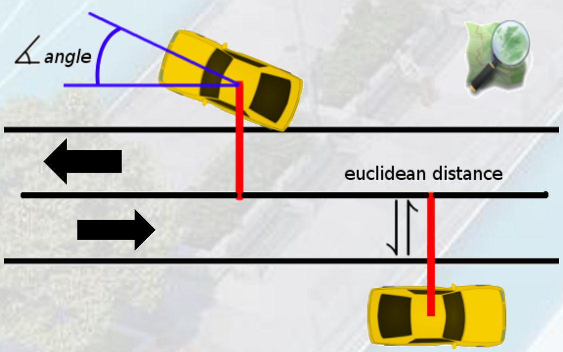


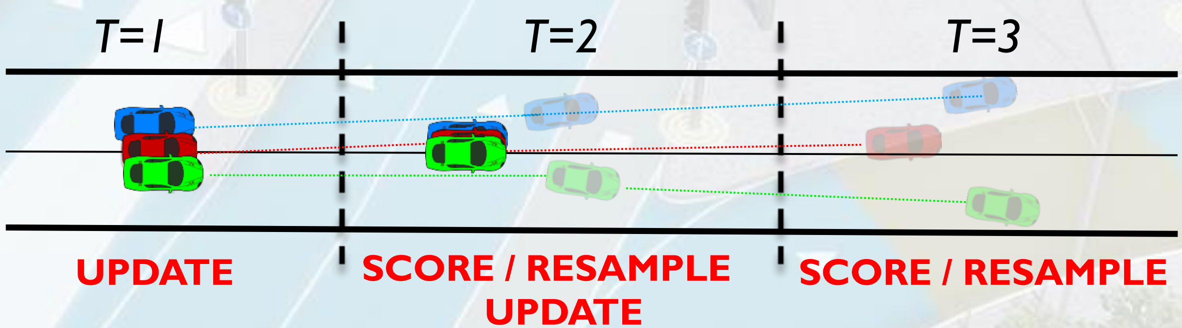
result of only three consecutive updates using
only the motion model

OpenStreetMap Component
3/3 Update Phase

The **OpenStreetMap Component** update routine allows us to weight hypotheses by means of the two following distance measures:

- Euclidean distance (m) from the nearest OSM road segment center
- Misalignment (rad) considering both the nearest OSM road segment center and driving direction tags





Please notice that *represents three localization hypotheses of the same vehicle*

OpenStreetMap Component
Experimental Activity

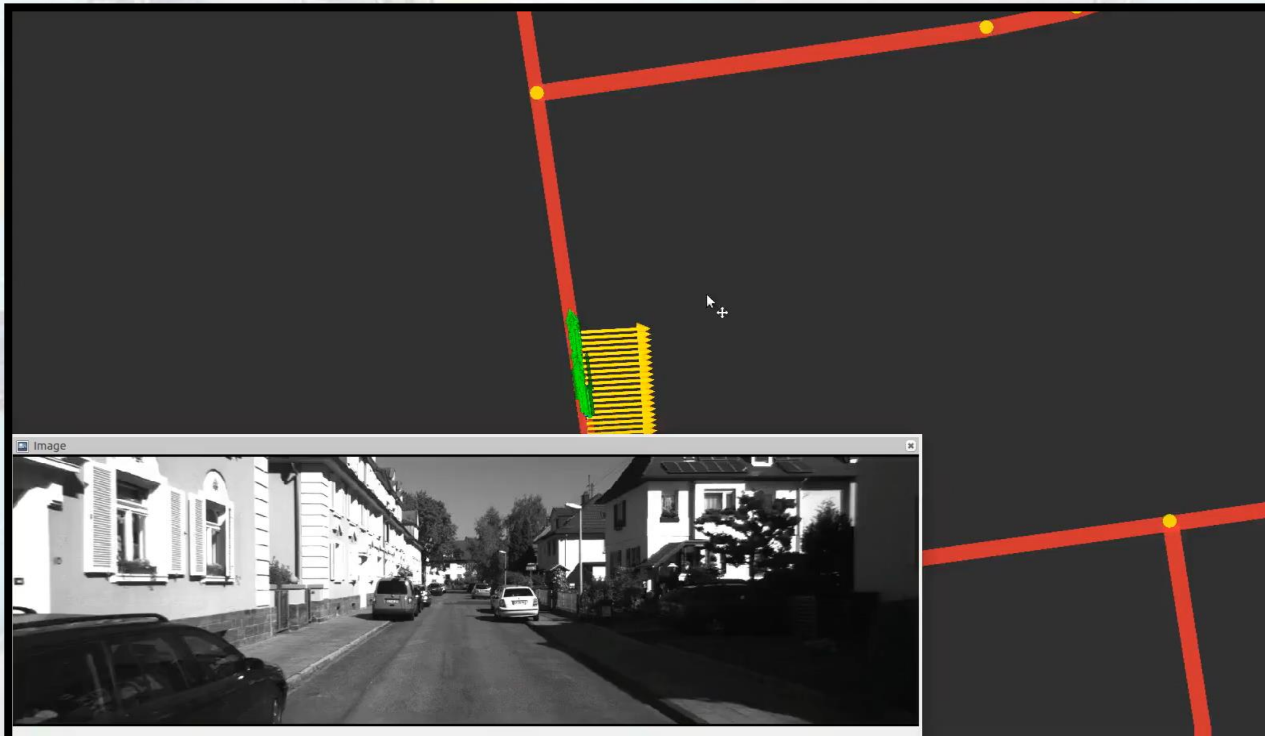
Approach tested on 10 sequences of the KITTI dataset

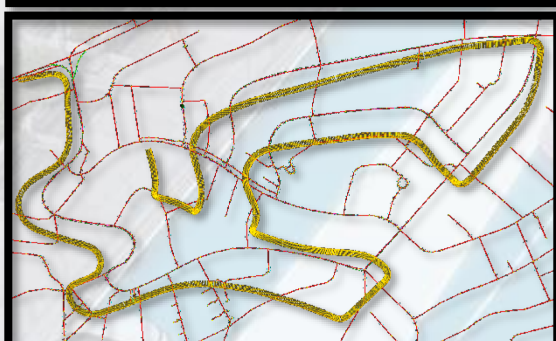
- Residential and Road categories
- An amplified initial uncertainty of 60m centered in the GPS fix
- RMSE error w.r.t. OpenStreetMap
- RMSE error w.r.t. Ground Truth (GPS-RTK)

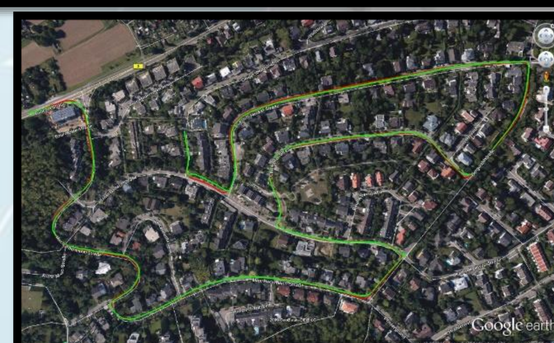
RLE + OpenStreetMap Component

A Framework for Outdoor Urban Environment Estimation

Intelligent Transportation Systems Conference **ITSC15**





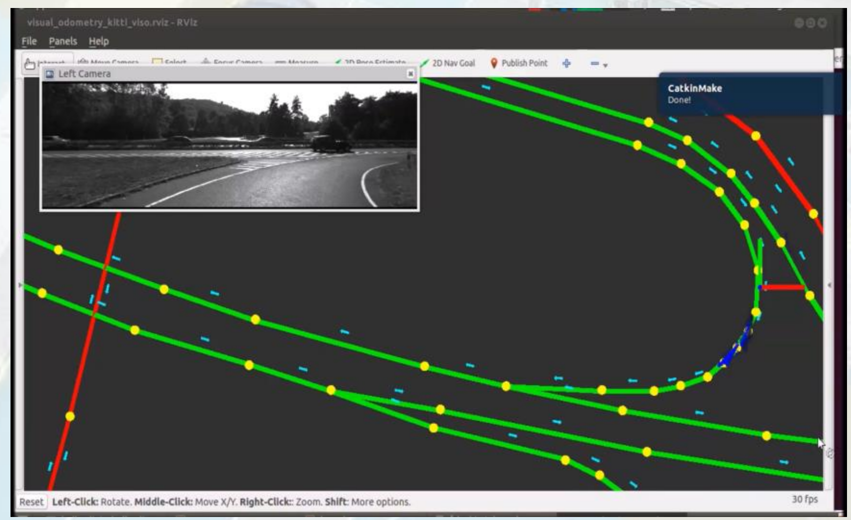


Performances

(benchmarked with respect to standard KITTI dataset)

RESULTS, 80 LAYOUT HYPOTHESES, AMPLIFIED GPS INITIALIZATION, OPENSTREETMAP + VISUAL ODOMETER (LIBVISO2)

	Sequence Length (mm:ss)	GPS-RTK Length (m) Used as Ground Truth	RLE RMSE (m) wrt OpenStreetMap	RLS RMSE (m) wrt GPS-RTK
TOTAL	38:38	19374,68		
AVERAGE	3:35	1937,46	$\boxed{0,936}$	$\boxed{1,688}$



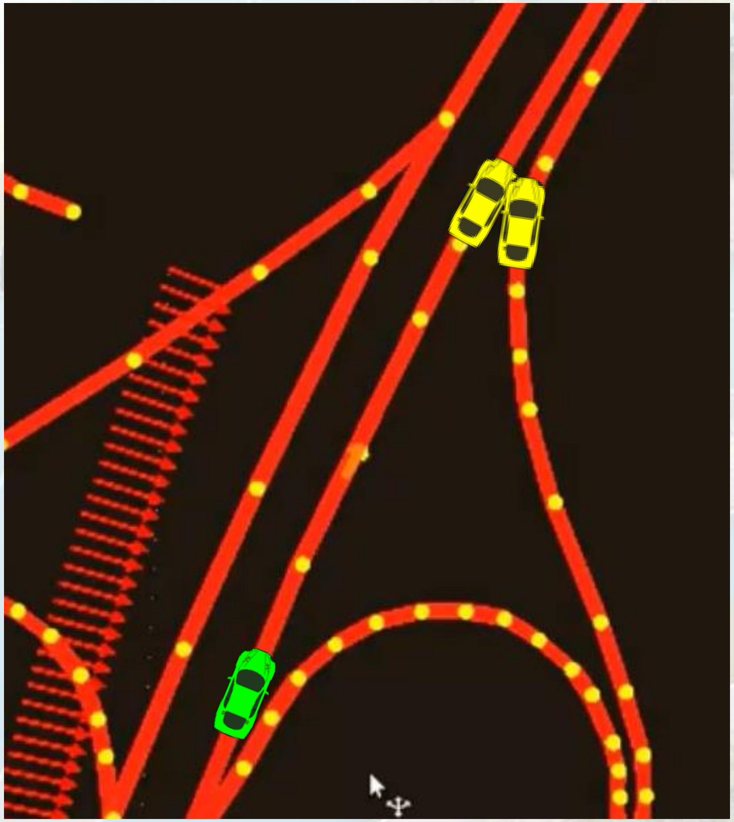
The Layout Managers runs at approx. 9,6Hz on a single threaded process, KITTI = 10Hz

OpenStreetMap Component
Does it always work?

Unfortunately, treacherous situations may arise …

- Initial Localization (green car)
  - Framework Estimates Positions (in time)
  - Localization Update (OpenStreetMap)

What Happens With two or more or Parallel Roads?



Leverage modularity to handle treacherous situations

Using ***only*** the OpenStreetMap Road segments introduce errors in the localization estimate both in

HIGHWAY SCENARIOS

and

URBAN AREAS

shared solution

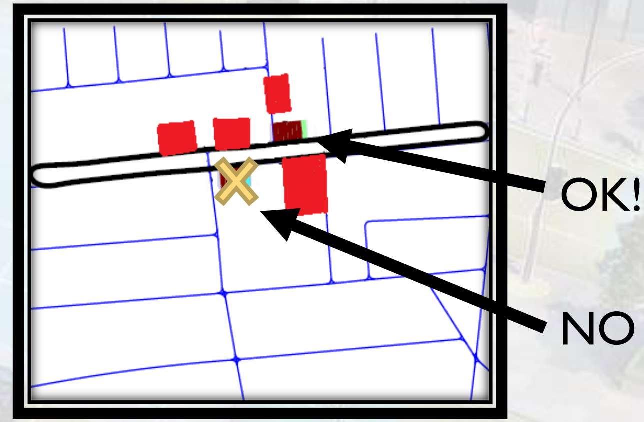
More Localization Components Needed

Green Points: Ground Truth Positions
Violet Points: Our Localization Estimate
Blue Line: OSM Road Segments

Which information to use? OpenStreetMaps Case Study

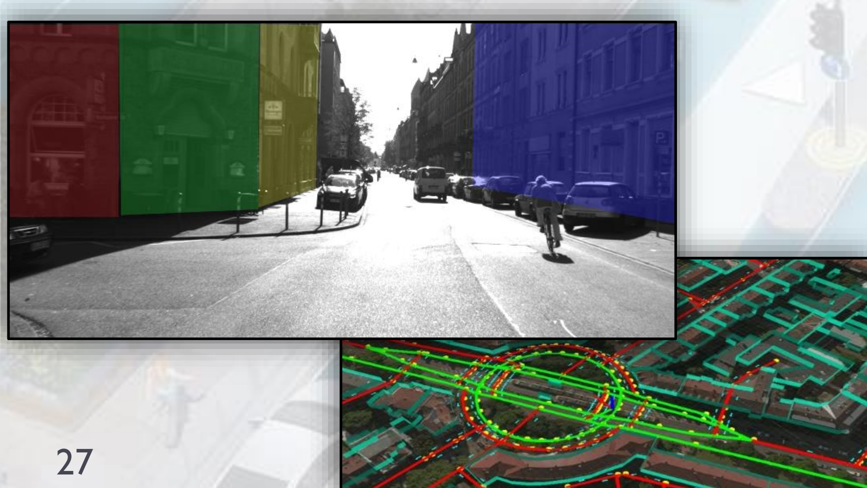
OpenStreetMaps has many useful features, including:

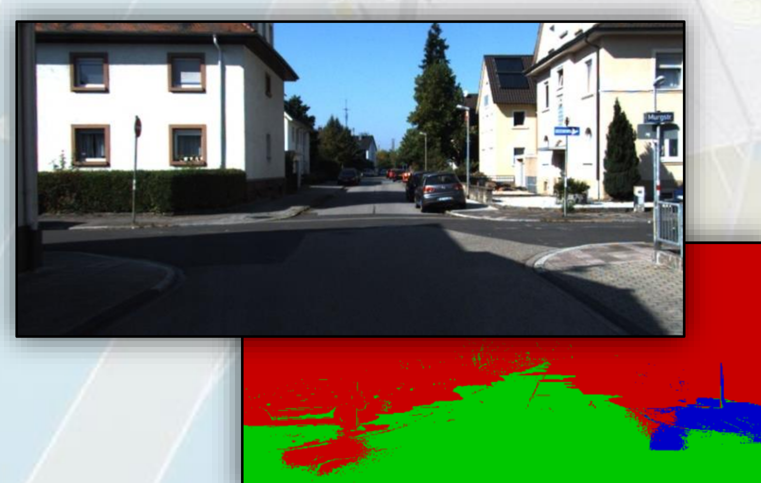
- **Buildings Outlines**
- Road driving directions
- **Roads lane numbers**
- Traffic signs and lights
- Barriers (curbs, guardrails…)



but we can also infer and exploit some "*higher semantic*" road features like:

- **Intersection Areas**





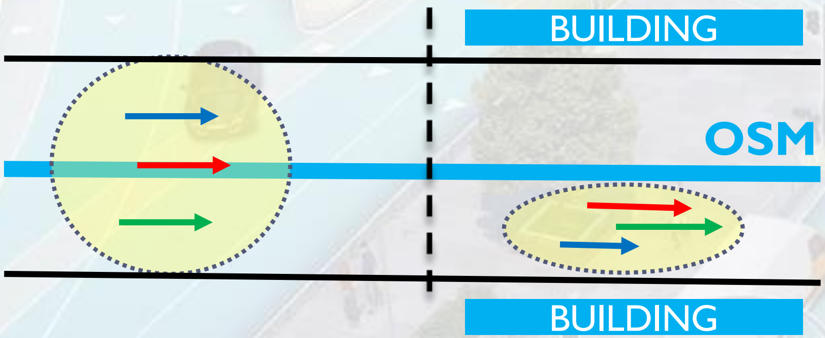
Leveraging Building Outlines

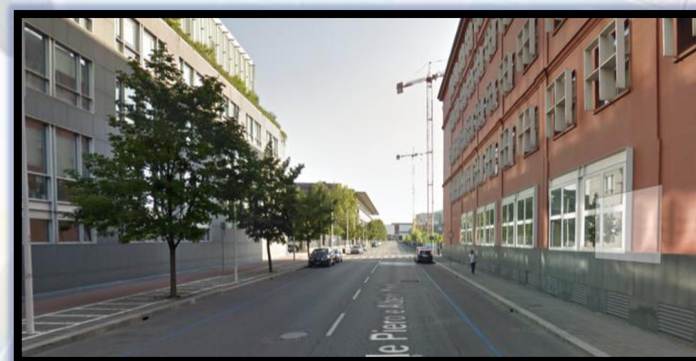
Why the Buildings?

To increase localization accuracy in challenging urban areas

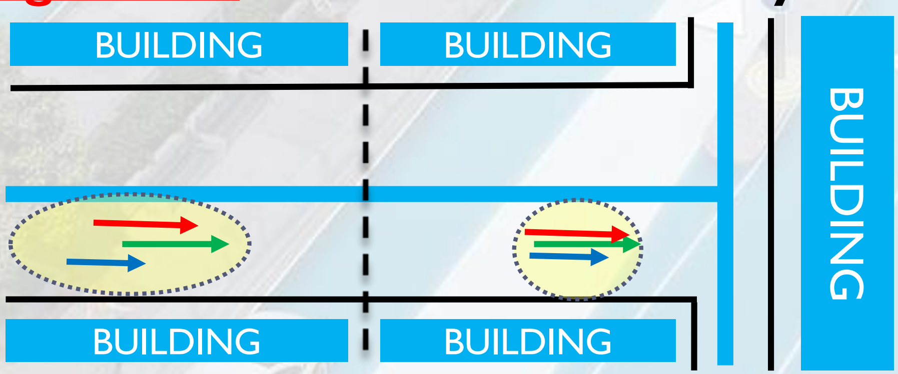
Lateral Localization Accuracy

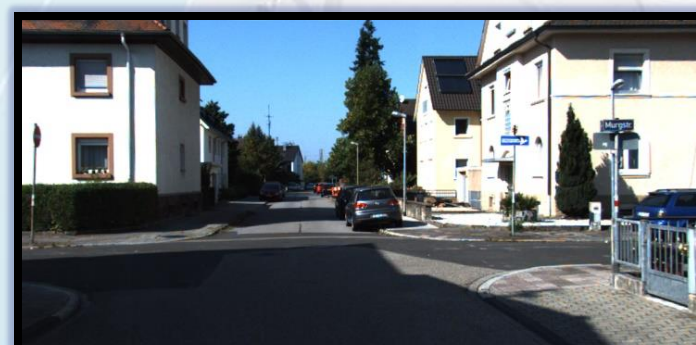
from ***On-Road*** localization to ***In-Lane*** localization





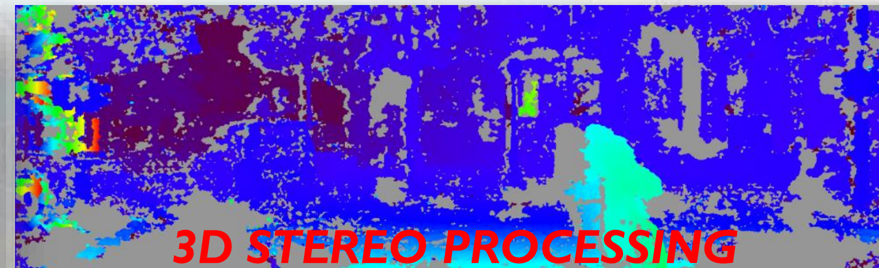
Longitudinal Localization Accuracy

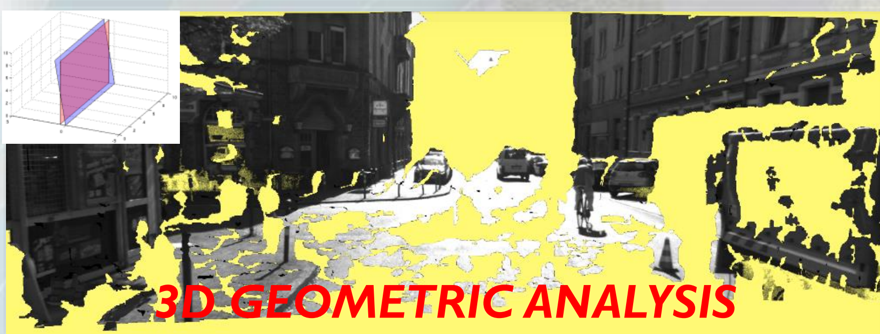


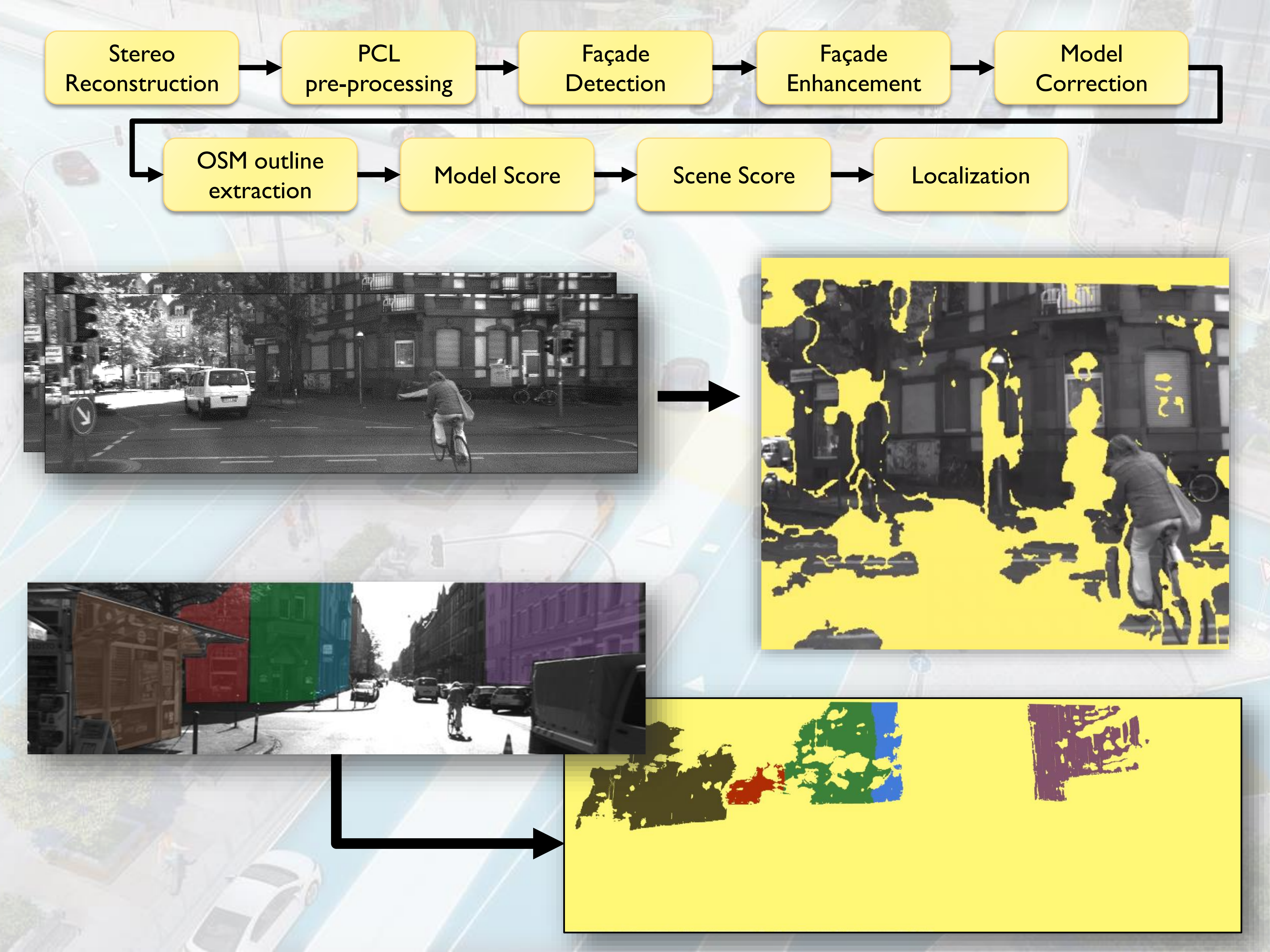


Leveraging Building Outlines

The detection pipeline involves the following steps



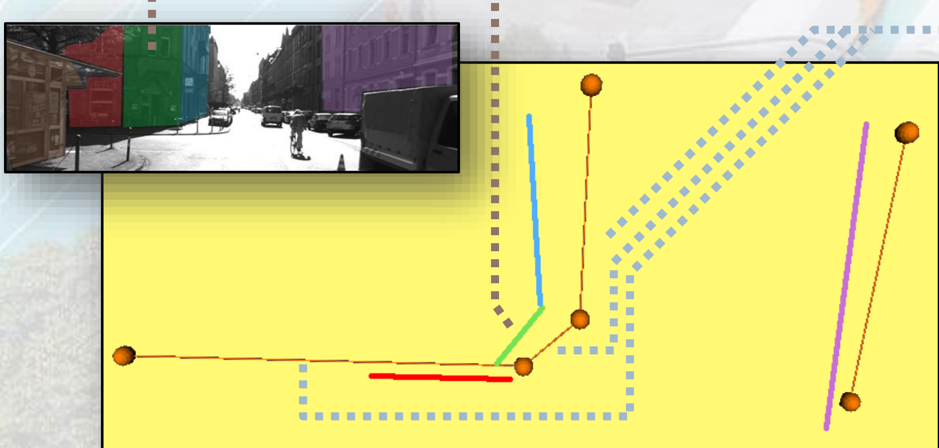



Stereo Reconstruction
PCL pre-processing
Façade Detection
Façade Enhancement
Model Correction
OSM outline extraction
Model Score
Scene Score
Localization

Stereo Reconstruction → PCL pre-processing → Façade Detection → Façade Enhancement → Model Correction → OSM outline extraction → Model Score → Scene Score → Localization

Layout Component Evaluation

Given a façade $f = < PCL_{Points}, \pi, CandidatePlanes, score: (F \times E) \rightarrow [0;1] >$



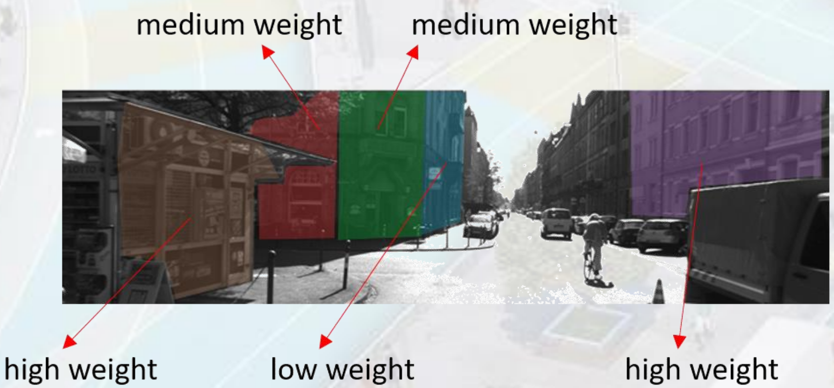
$$\bar{d} = \frac{1}{|P|} \sum_{i=1}^{|P|} \frac{|a_\epsilon x_i + b_\epsilon y_i + c_\epsilon z_i + d_\epsilon|}{\sqrt{a_\epsilon^2 + b_\epsilon^2 + c_\epsilon^2}}$$

Average distance between inliers (PCL_points) and the OSM Outline

$$\alpha = \arccos\left(\frac{a_\pi a_\epsilon + b_\pi b_\epsilon + c_\pi c_\epsilon}{\sqrt{a_\pi^2 + b_\pi^2 + c_\pi^2}\sqrt{a_\epsilon^2 + b_\epsilon^2 + c_\epsilon^2}}\right)$$

Angular distance betweend the corrected perpendicular model and the OSM Outline

Stereo Reconstruction → PCL pre-processing → Façade Detection → Façade Enhancement → Model Correction → OSM outline extraction → Model Score → Scene Score → Localization

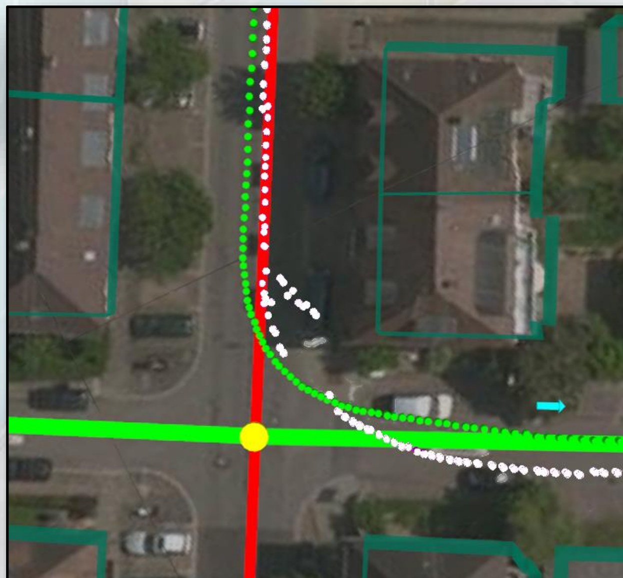


$$scene\_score = \frac{\sum_{i=1}^{F} |P_i| \cdot score(f_i)}{\sum_{i=1}^{F} |P_i|}$$

Comparison with Previous Work 1/2

Localization Without Building Matching

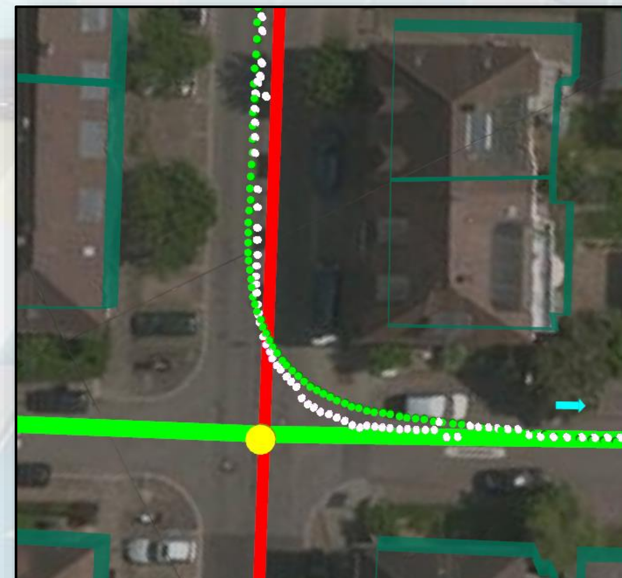
A Framework for Outdoor Urban Environment Estimation
Ballardini et. al. - Intelligent Transportation Systems (ITSC) 2015



Green dots: RTK GPS used ad Ground Truth
White dots: Evaluated position (Localization)

Localization With Building Matching

Leveraging the OSM Building Data to Enhance the Localization of an Urban Vehicle, Ballardini et. al. - Intelligent Transportation Systems (ITSC) 2016



Red Line: OpenStreetMap Road
Red Green: OpenStreetMap Road (oneway)

Comparison with Previous Work 2/2

Localization Without **Building Matching**

A Framework for Outdoor Urban Environment Estimation
Ballardini et. al. - Intelligent Transportation Systems (ITSC) 2015

Green dots: RTK GPS used ad Ground Truth
White dots: Evaluated position (Localization)

Localization With **Building Matching**

Leveraging the OSM Building Data to Enhance the Localization of an Urban Vehicle, Ballardini et. al. - Intelligent Transportation Systems (ITSC) 2016

Red Line: OpenStreetMap Road
Red Green: OpenStreetMap Road (oneway)

Performance Gain

with «buildings enabled»

Building Detector Pipeline

Sequence Name	Sequence	Category	GPS-RTK (m)
2011_10_03_drive_0027	7:50	Residential	2,651.92
2011_10_03_drive_0034	8:03	Residential	2,872.90
2011_09_30_drive_0018	4:47	Residential	2,205.77
2011_09_30_drive_0020	1:53	Residential	1,227.57
2011_09_30_drive_0027	1:53	Residential	693.12
2011_09_30_drive_0028	7:02	Residential	3,204.46
2011_09_30_drive_0033	2:44	Residential	1,700.71
2011_09_30_drive_0034	2:04	Residential	918.99
2011_09_26_drive_0005	0:16	Residential	66,10
2011_09_26_drive_0046	0:13	Residential	46.38
2011_09_26_drive_0095	0:27	Residential	252.63
TOTAL	36:16		15,475.44

Sequence Name	Original RMSE	Our Proposal
2011_09_26_drive_0005	2.52716	1.92298
2011_09_26_drive_0046	2.40916	1.64384
2011_09_26_drive_0095	2.66319	1.47326

Conclusions

- We achieved <u>**Lane Level**</u> Localization in urban environments
- First step towards OpenStreetMaps feature integration/exploitation
- Good enhancement over *state of the art* road localization algorithms relying on *lock-on-road* procedures

IEEE ITSC
RIO 2016

OpenStreetMap Component
+
Building Component

Does it always work?

Unfortunately, treacherous situations may still arise …

Areas without Buildings

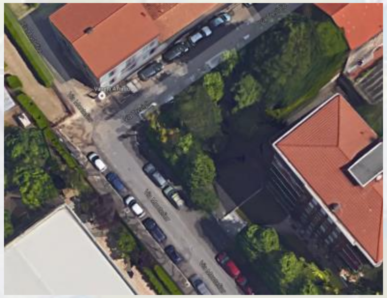
Introducing *«High Level Features»*: *Detecting The Road Intersections*

…using the *buildings* as anchor points is good, but they are not always available…

how to further reduce longitudinal uncertainties?

We can go beyond OpenStreetMaps features introducing
*“**high level**”* features **<u>derived</u>** from basic OpenStreetMap features, *i.e.*, **roads**



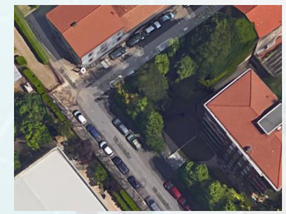



from ***<u>ROAD SEGMENTS</u>*** to ***<u>ROAD INTERSECTIONS</u>***

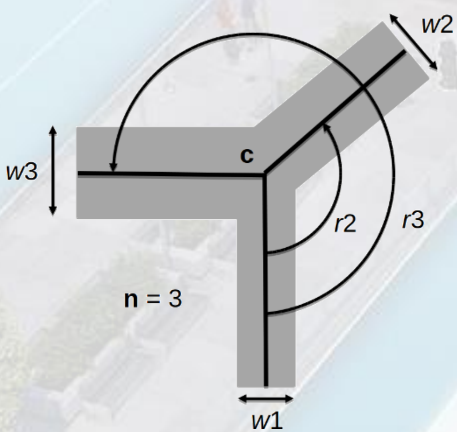
Introducing *«High Level Features»*
Detecting The Road Intersections

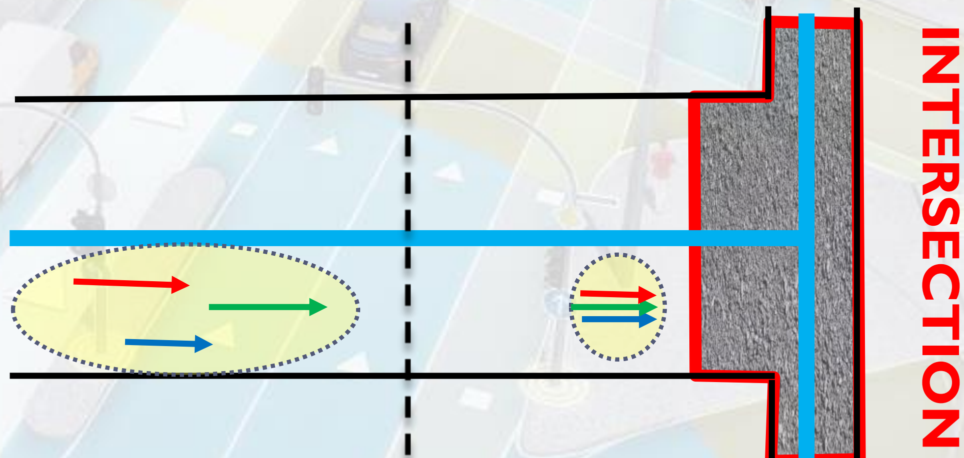
We can go beyond OpenStreetMaps features introducing
"higher semantic" features derived from base OpenStreetMap features



Intersection model



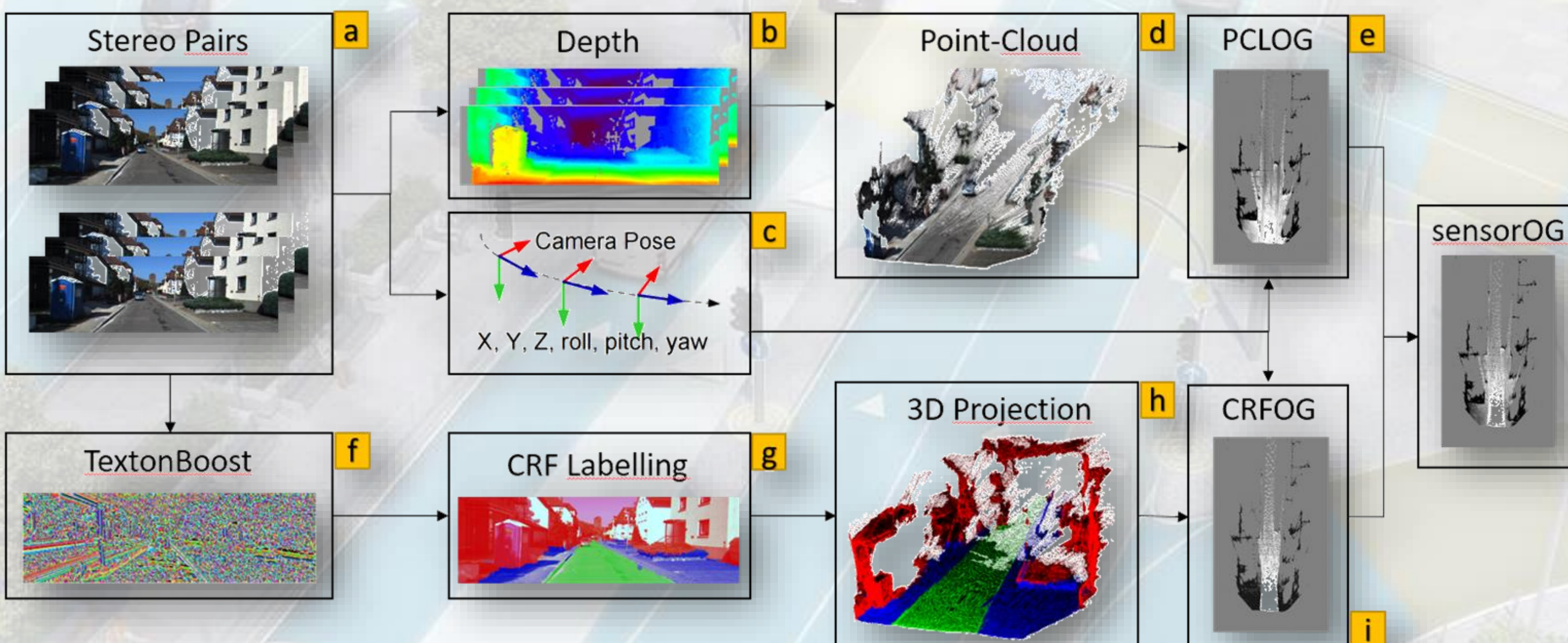


N = number of approaching arms
C = intersection center position
w_n= with of the road segment
r_n = rotation with respect to the current road segment

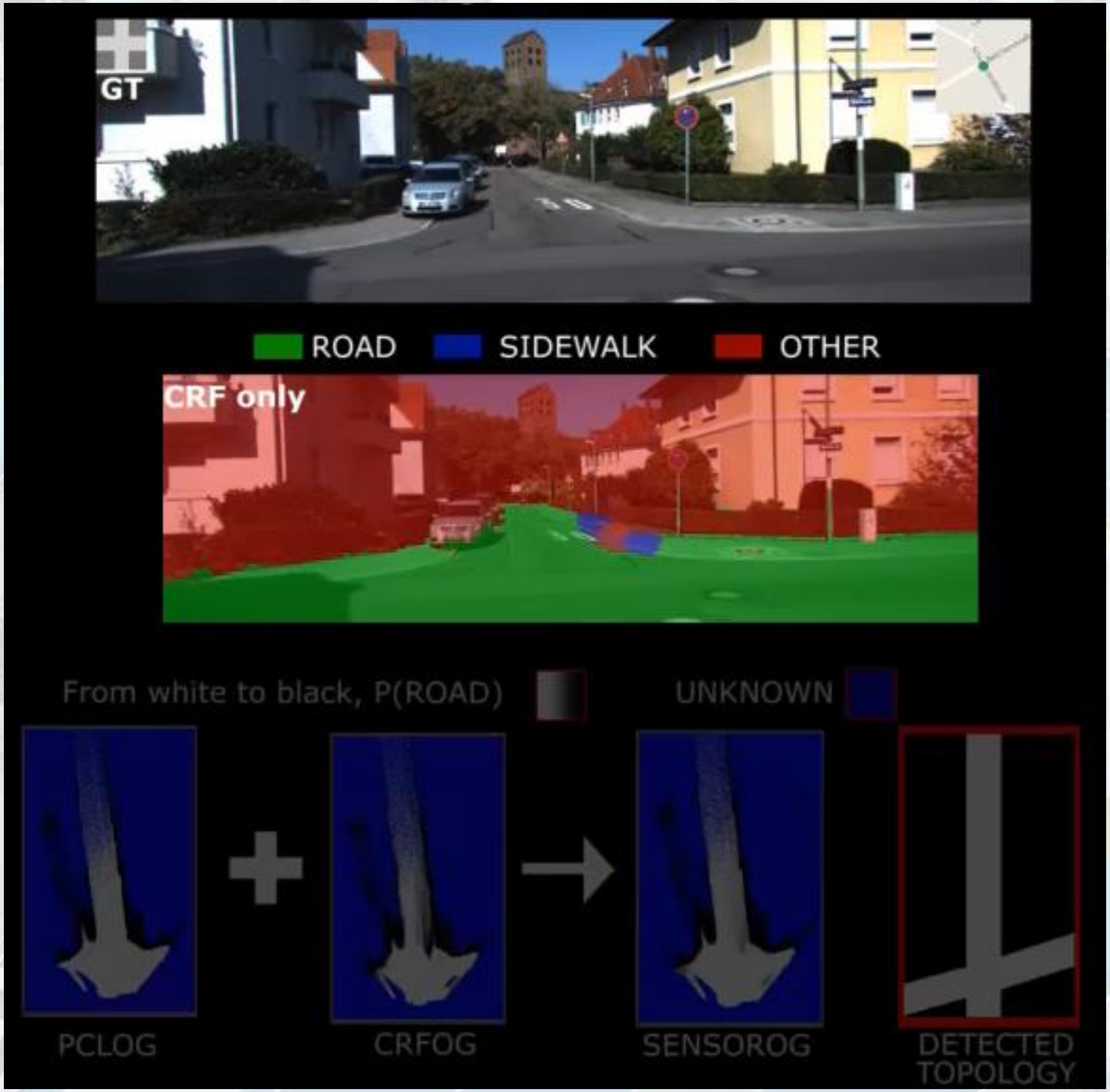
Introducing *«High Level Features»* *Detecting The Road Intersections*

The Intersection detection pipeline involves a pixel-wise classification of an image captured from a moving vehicle, by means of a synergically exploitation of the following two approaches:

1. **Image analysis, using a Conditional Random Field approach**
2. **3D points evaluation, retrieved from stereo images**



Introducing *«High Level Features»*
Detecting The Road Intersections



Introducing *«High Level Features»*
Detecting The Road Intersections

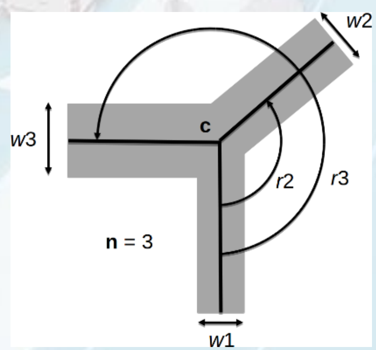
Detecting the intersection in urban areas is hard due to <u>frequent presence of strong clutter</u>

The proposed detector allows us to generate an evaluation of the perceived scene in "*road - topological*" terms. The proposed approach achieved better results with respect to the literature approaches.

Intersection Topology Identification

	0.91						
	0.25	0.62	0.06				
	0.29	0.09	0.41				0.11
	0.66	0.15	0.07	0.07			
	0.47	0.21			0.14		0.11
	0.14	0.19	0.54				0.10
	0.59		0.22				0.13

Ess. et al. BMVC 2009



	0.75	0.04	0.04	0.06	0.11	0.00	0.01	12088
	0.39	0.32	0.11	0.01	0.14	0.01	0.04	338
	0.40	0.10	0.37	0.05	0.06	0.02	0.00	241
	0.36	0.15	0.06	0.29	0.09	0.00	0.05	722
	0.38	0.07	0.04	0.12	0.33	0.01	0.05	689
	0.36	0.21	0.23	0.02	0.00	0.14	0.04	826
	0.25	0.07	0.04	0.10	0.19	0.02	0.33	1673

Our approach

ICRA2017
May 29 – June 3, 2017 • Singapore

OpenStreetMap Component
+
Building Component
+
Road Intersection Component

Does it always work?

Unfortunately, treacherous situations may arise …

Urban, but
No Buildings, No Intersections

Highways

A different kind of components …

Urban Wide Avenues and Highway areas do ***not*** *have* ***BUILDINGS*** *or* ***INTERSECTION*** *areas but additional information is still needed to perform a good localization*

The flexibility of the framework allows to perform the localization adding a new Layout Component

Example
Issues in Highway Scenarios

Typical localization errors arise near highway ramps at tracks merge points, or in case of parallel roads

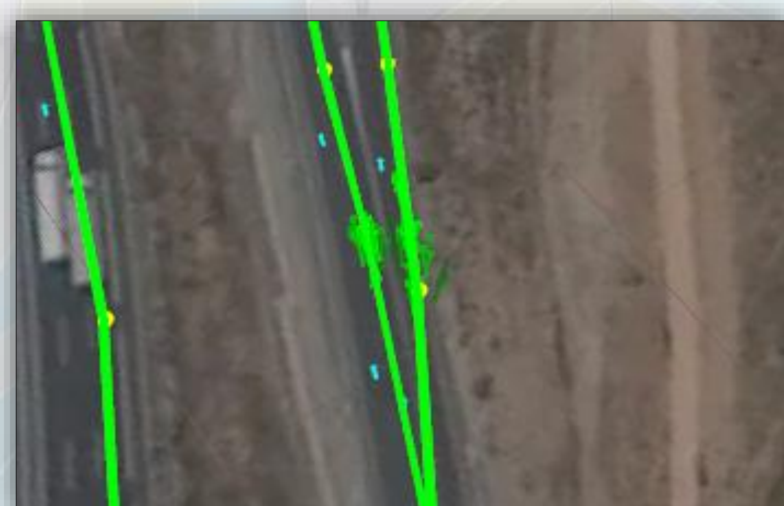
Line Detector & Tracker

in collaboration with ISIS Lab, University of Alcalá - Spain

\+

OpenStreetMap

- *Road Width (m)*
- *Number of Lanes*



Road Width component

discriminate different parallel roads

Road Lanes component

achieve in lane localization

Results using Road Width Component

Qualitative Results

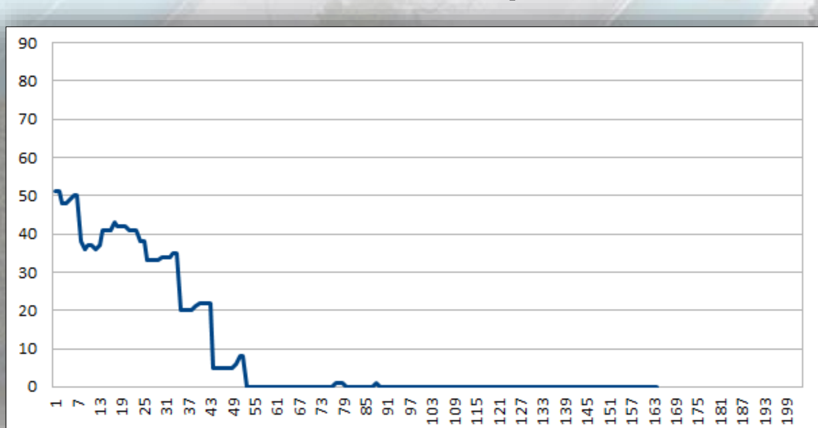
Road Width Disabled

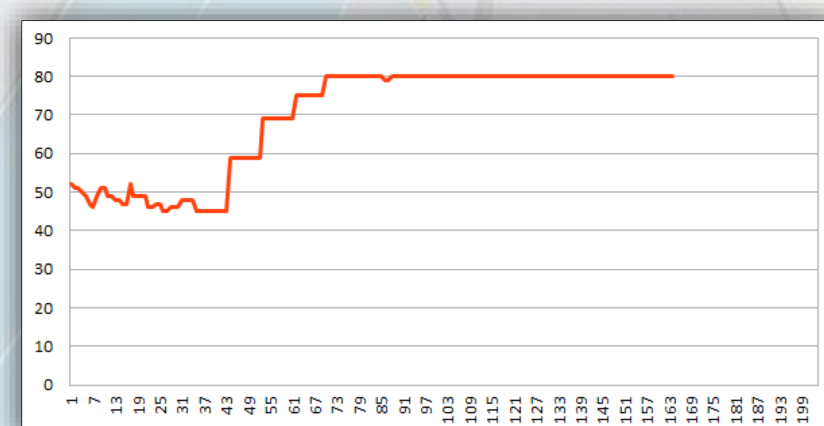
Road Width Enabled

Results using Road Width Component

Quantitative Results

(number of localized particles)

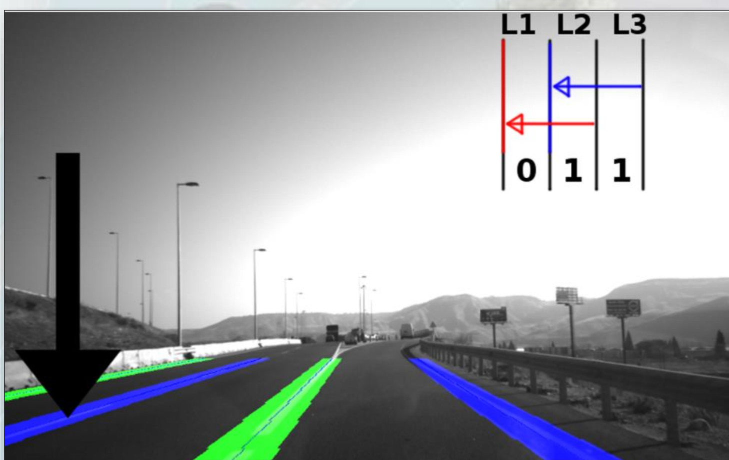




In the shown experiment, the frame rate was set to 10 frames per second and the total number of hypotheses was set to 80

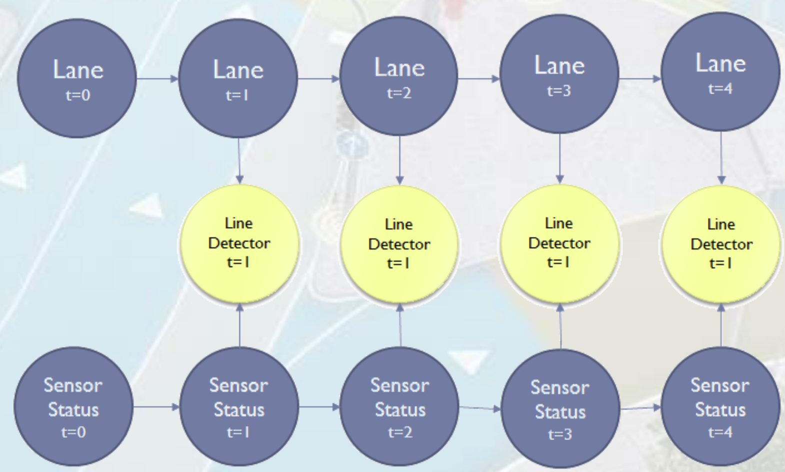
Road Lane Component

- Identifies the current number of lanes (from OpenStreetMap) and the Vehicle Ego-Lane
- Track the Vehicle Ego-Lane in Time by means of a HMM Approach



Considering the line indicated with the arrow, the probability of being in Lane$_{\{1|2|3\}}$ is estimated as {0,0.33,0.33}

Hidden Markov Model With Transient Failure Model



Road Lane Component

- Identifies the current number of lanes (from OpenStreetMap) and the Vehicle Ego-Lane
- Track the Vehicle Ego-Lane in Time by means of a HMM Approach

Quantitative Results with respect to the Ego-Lane detected with the Sensor Only

Table 5.9: HMM vs Naive Detector Ego-Lane Estimate

	Lane 1	Lane 2	Lane 3	Sum of Errors	Fault Rate
Detector Failures	91	216	25	364	0.70
HMM Failures	37	75	76	188	0.36
Ground Truth frames in lines	219	216	82		

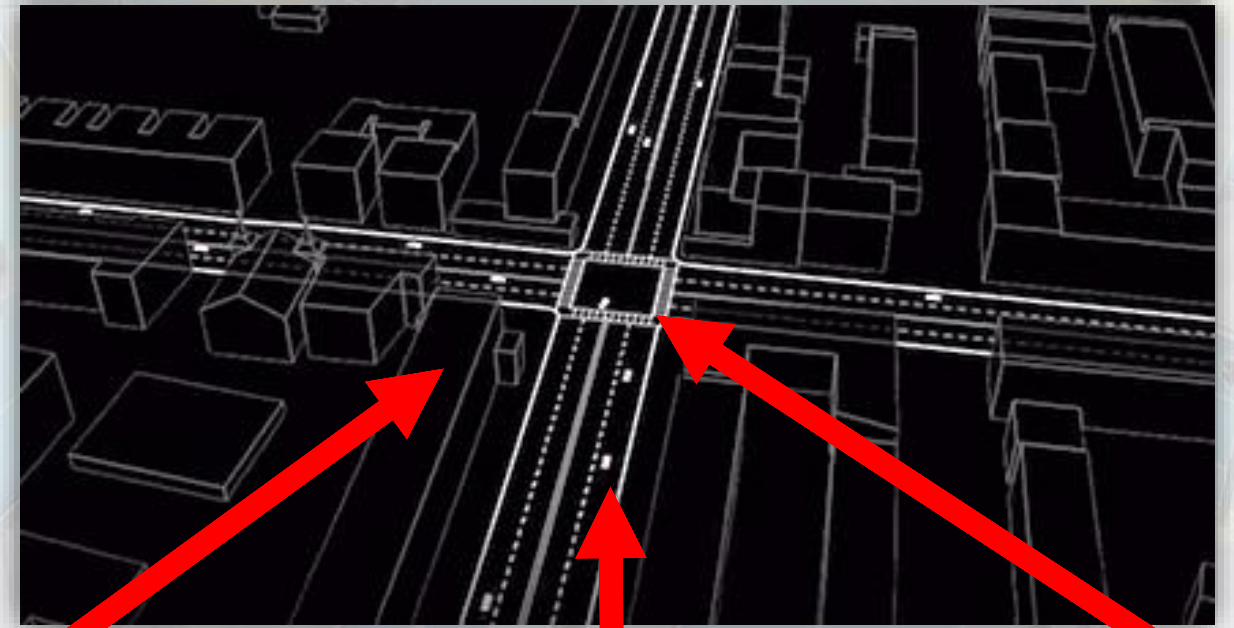
Conclusions

We have presented a probabilistic framework aimed at estimating the ego-vehicle localization in both urban and highway scenarios.

- Introduced the "Road Layout Estimation" Framework
  - Integrated with OpenStreetMap
  - Leverages the existent Road Network for lock-on-road
  - Exploits the Building's outlines for in-lane localization
  - Uses Road Width and Lanes to reduce uncertainties
- Detected and classified the Road Intersections

Future Works

The RLE framework opens a new set of research challenges



OTHER STATIC OBJECTS

INTEGRATION OF MOVING OBJECTS

COMPONENTS INTERACTIVITY

Why LOCALIZATION is important?☺

Thank you
Q&A

Matching Heterogeneous Sensing Pipelines to Digital Maps For Ego-Vehicle Localization

Thank you,

question time

Augusto Luis Ballardini
ballardini@disco.unimib.it