



HAL
open science

Dynamic Traffic Scene Understanding using Bayesian Sensor Fusion and Motion Prediction

Christian Laugier

► **To cite this version:**

Christian Laugier. Dynamic Traffic Scene Understanding using Bayesian Sensor Fusion and Motion Prediction. ECCV 2018 - Workshop on Vision-based Navigation for Autonomous Driving, Sep 2018, Munich, Germany. pp.1-35. hal-01968786

HAL Id: hal-01968786

<https://inria.hal.science/hal-01968786v1>

Submitted on 4 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Dynamic Traffic Scene Understanding using Bayesian Sensor Fusion and Motion Prediction

Christian LAUGIER, Research Director at Inria

*Inria Chroma team & IRT nanoelec & Scientific Advisor for Probayes and for Baidu China
christian.laugier@inria.fr*

Contributions from

*Lukas Rummelhard, Amaury Negre, Nicolas Turro, Jean-Alix David, Jérôme Lussereau,
Christopher Tay Meng Keat, Stephanie Lefevre, Javier-Ibanez Guzman*



ADAS & Autonomous Driving

Invited Talk (40mn)

ECCV2018 Workshop ApolloScape “Vision-based Navigation for Autonomous Driving”

Gasteig Cultural Center, Rosenheimer Str.5, Munich, September 9th 2018

Autonomous Cars & Driverless Vehicles

- Strong involvement of Car Industry & Large media coverage
- An expected market of 500 B€ in 2035
- Technologies Validation => Numerous recent & on going real-life experiments (in addition to simulation & formal methods)



Tesla Autopilot based on Radar & Mobileye



3D Lidars & Dense 3D mapping
Numerous vehicles & Millions of miles driven



Mobility Service
Cybus experiment, La Rochelle 2012
=> CityMobil Project & Inria



Drive Me trials (Volvo, 2017)

- 100 Test Vehicles in Göteborg, 80 km, 70km/h
- No pedestrians & Plenty of separations between lanes



Robot Taxi testing in US (Uber, Waymo) & Singapore (nuTonomy)
=> **Mobility Service**, Numerous Sensors ... Safety driver in the car during testing



Millions of miles driven (Tesla, Waymo, Uber...)
Several benign & serious accidents in past few years!
Safety is still not guaranteed!

Safety issues: *Example of the Tesla accident (May 2016)*

❑ Safety is still insufficient (*a false sense of Safety for users ?*)

=> Still some Perception & Situation Awareness errors (even with commercial systems)

=> On May 7th 2016, Tesla driver killed in a crash with Autopilot active (and driver not attentive)

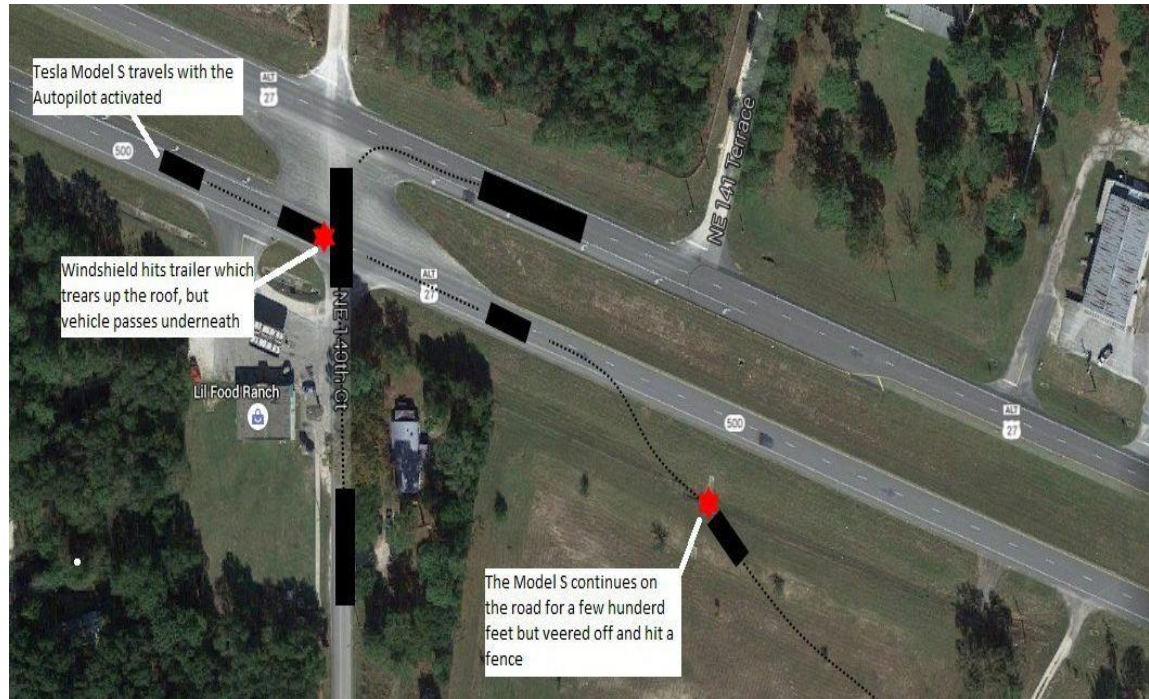


Displayed information

Tesla Model S – Autopilot

Front perception:

Camera (Mobileye) + Radar + US sensors



Autopilot didn't detected the trailer as an obstacle (NHTSA investigation + Tesla conjecture)

❖ **Camera** => *White color against a brightly lit sky ?*

❖ **Radar** => *High ride height of the trailer probably confused the radar into thinking it is an overhead road sign ?*

Safety issues: *Example of the Uber Accident (March 2018)*

- ❑ **Self-driving Uber kills a woman in first fatal crash involving pedestrian**
Tempe, Arizona, March 2018
- ❑ **The vehicle was moving at 40 mph and didn't reduced its speed before the crash (collision risk not detected). The Safety Driver didn't reacted**
- ❑ **In spite of the presence of multiple onboard sensors (several lidars in particular), the perception system didn't predicted the collision !**

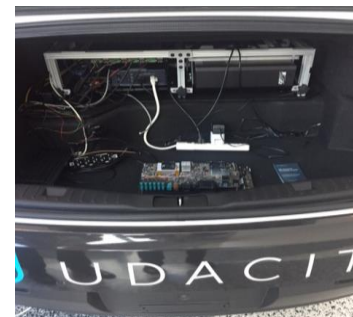


Perception: State of the Art & Today's Limitations

- ❑ Despite significant improvements during the last decade of both Sensors & Algorithms, **Embedded Perception** is still one of the major bottleneck for Motion Autonomy

=> Obstacles detection & classification errors, incomplete processing of mobile obstacles, collision risk weakly address, scene understanding partly solved...

- ❑ **Lack of Robustness & Efficiency & Embedded integration** is still a significant obstacle to a full deployment of these technologies



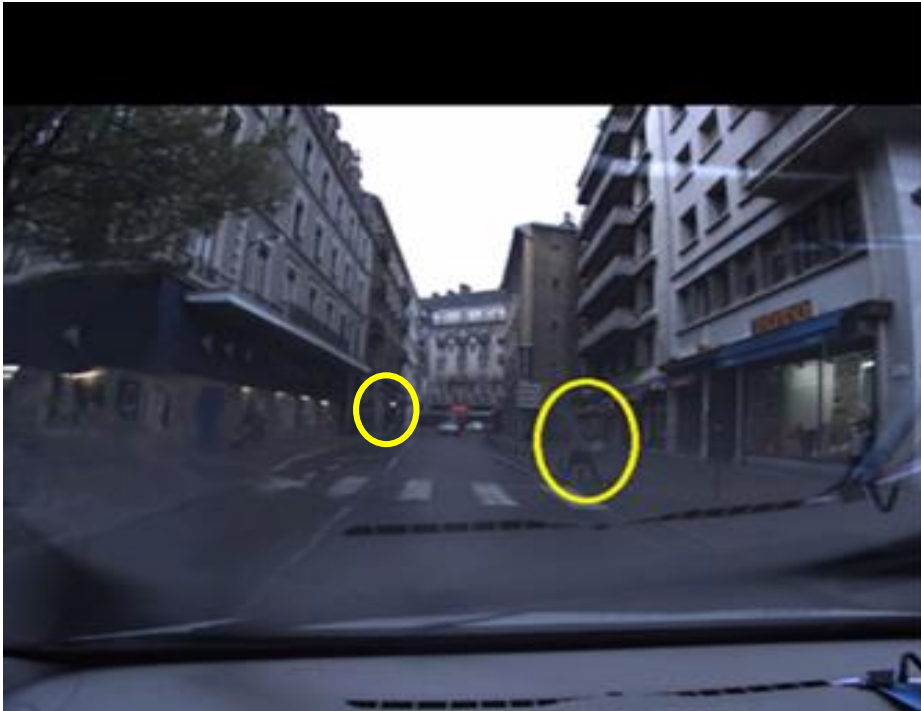
Lack of Robustness & Efficiency ✘

Lack of Integration into Embedded Sw/Hw ✘

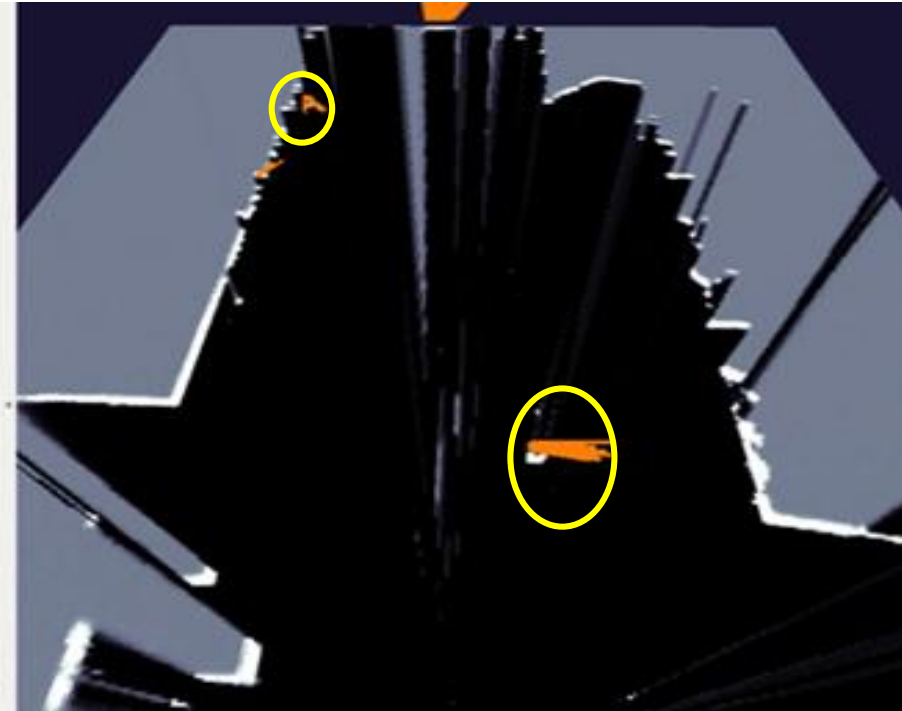
- Until recently, car trunks was most of the time full of electronics & computers & processor units
- On-board high computational capabilities & dedicated softwares are still required, even if new products currently appear on the market (e.g. Nvidia Drive-PX, Ambarella embedded vision platform ..)

Improving robustness using Multi-Sensors Fusion

Camera Image at Dusk (*Pedestrians not detected*)



Processed Lidar data (*Pedestrians detected*)



Camera output depends on lighting conditions
Cheap & Rich information & Good for classification

Lidar more accurate & can work at night
Good for fine detection of objects ... but still Expensive

- *Any sensor may generate perception errors ! It is mandatory to develop **Embedded Robust & Efficient Multi-Sensors Fusion** approaches (e.g. using probabilistic models & filtering)*
- *A new generation of **affordable “Solid State Lidars”** is supposed to shortly arrive on the market !*
 - => *No mechanical component & Expected cost less than 1000 US\$ before mass production*
 - => *Numerous announcements since Spring 2016 ... but products not yet on the market !!*



Focus of the talk

Complex Dynamic Scenes
understanding



**Situation Awareness
& Decision-making**



ADAS & Autonomous Driving

Dealing with unexpected events
e.g. Road Safety Campaign, France 2014



Anticipation & Prediction
for avoiding upcoming accidents

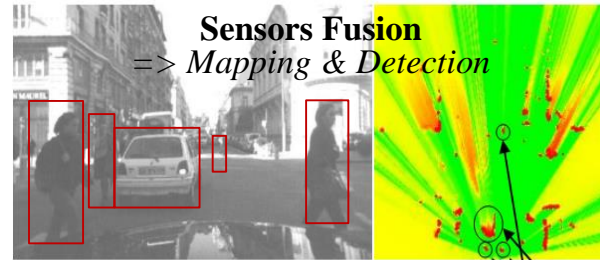
Main features

- ✓ Dynamic & Open Environments => *Real-time processing*
- ✓ Incompleteness & Uncertainty => *Appropriate Model & Algorithms (probabilistic approaches)*
- ✓ Sensors limitations => *Multi-Sensors Fusion*
- ✓ Human in the loop => *Interaction & Behaviors & Social Constraints (including traffic rules)*
- ✓ Hardware / Software integration => *Satisfying Embedded constraints*

Key Technology 1: Embedded Bayesian Perception



Embedded Multi-Sensors Perception
⇒ *Continuous monitoring of the dynamic environment*



❑ Main challenges

- ✓ *Noisy data, Incompleteness, Dynamicity, Discrete measurements*
- ✓ *Strong Embedded & Real time constraints*

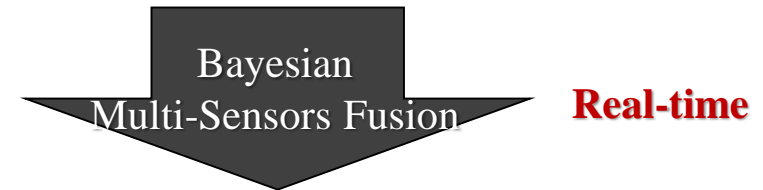
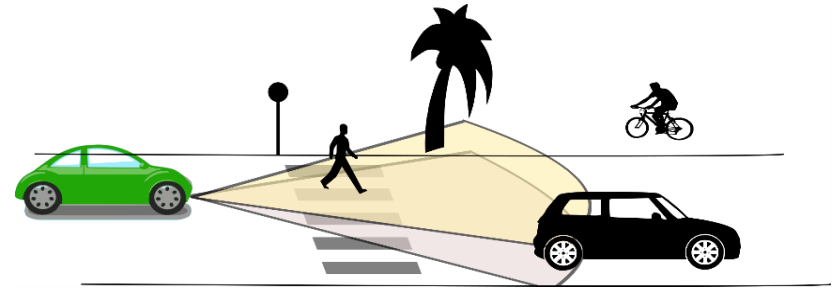
❑ Our Approach: Embedded Bayesian Perception

- ✓ *Reasoning about Uncertainty & Time window (Past & Future events)*
- ✓ *Improving robustness using Bayesian Sensors Fusion*
- ✓ *Interpreting the dynamic scene using Contextual & Semantic information*
- ✓ *Software & Hardware integration using GPU, Multicore, Microcontrollers...*

Bayesian Perception : Basic idea


□ Multi-Sensors Observations

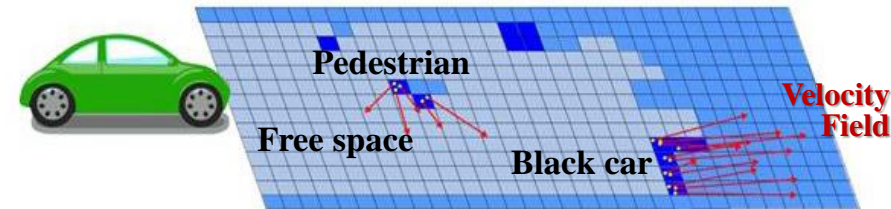
Lidar, Radar, Stereo camera, IMU ...



□ Probabilistic Environment Model

- ✓ *Sensor Fusion*
- ✓ *Occupancy grid integrating uncertainty*
- ✓ *Probabilistic representation of Velocities*
- ✓ *Prediction models*

$P[o|Z,C]$: 



Concept of Dynamic Probabilistic Grid
⇒ Occupancy & Velocity probabilities
⇒ Embedded models for Motion Prediction

□ Main philosophy

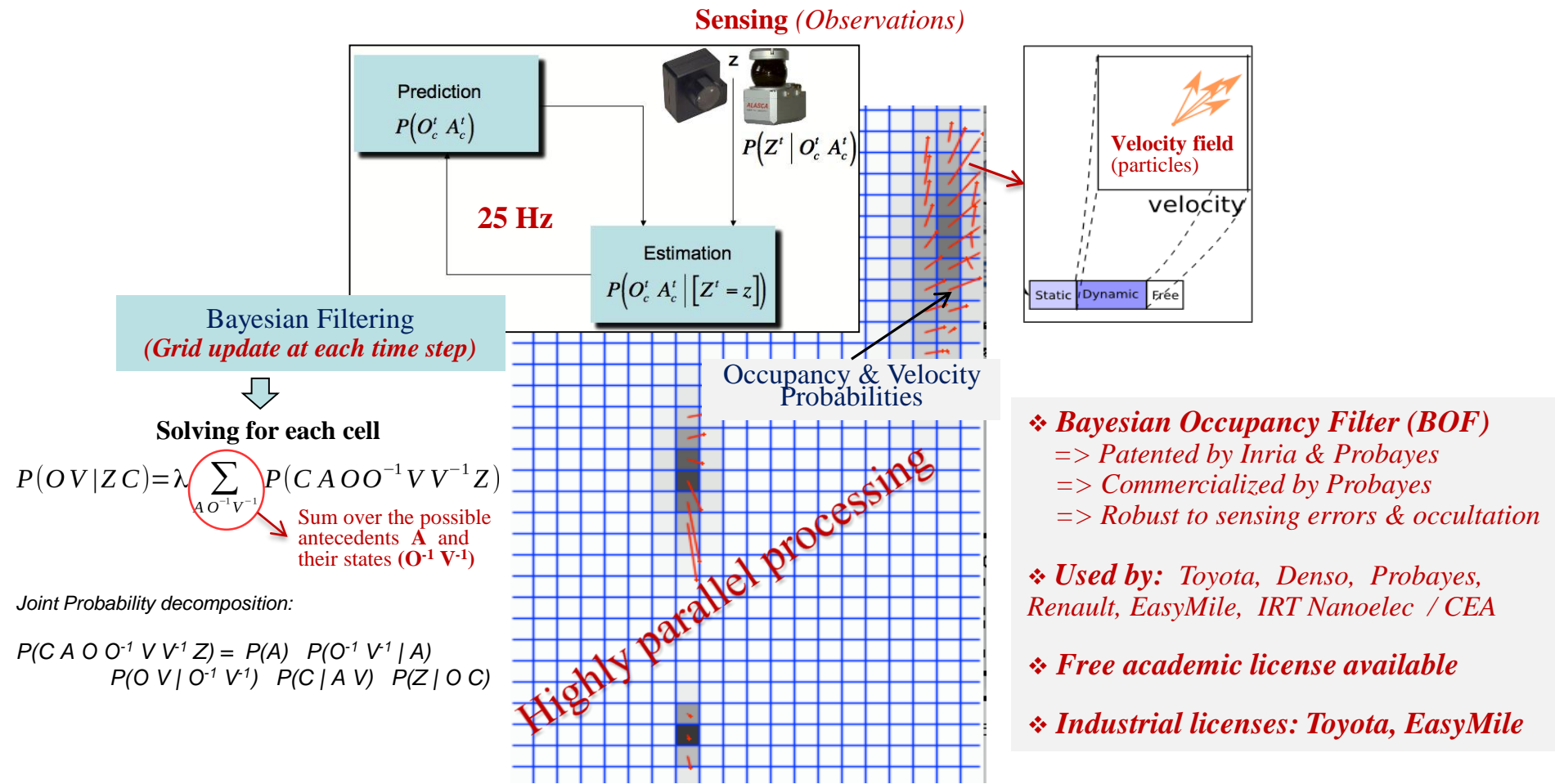
Reasoning at the grid level as far as possible for both :

- **Improving efficiency** (highly parallel processing)
- **Avoiding traditional object level processing problems** (e.g. detection errors, wrong data association...)

A new framework: Dynamic Probabilistic Grids

=> A clear distinction between Static & Dynamic & Free components

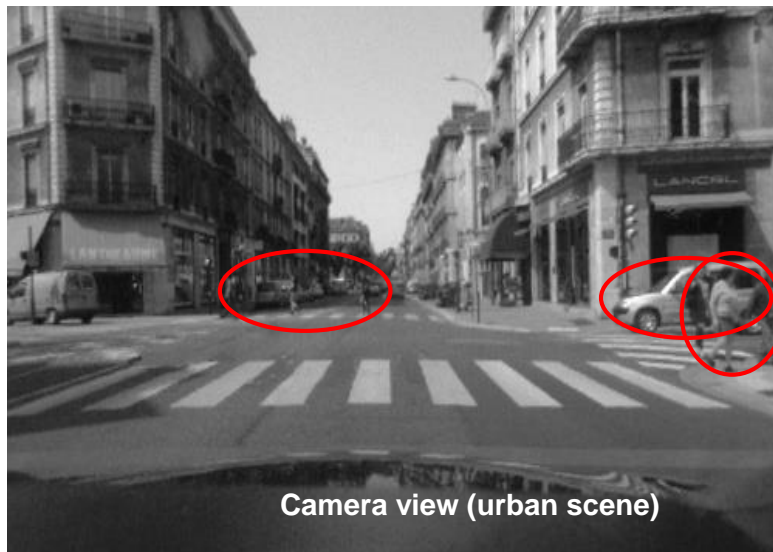
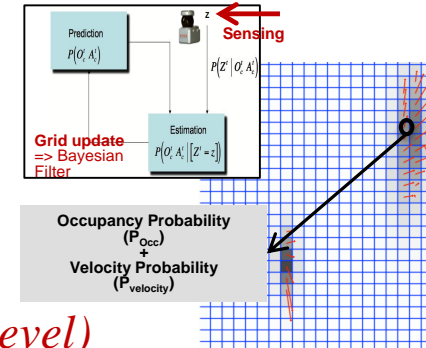
[Coué & Laugier IJRR 05] [Laugier et al ITSM 2011] [Laugier, Vasquez, Martinelli Mooc uTOP 2015]



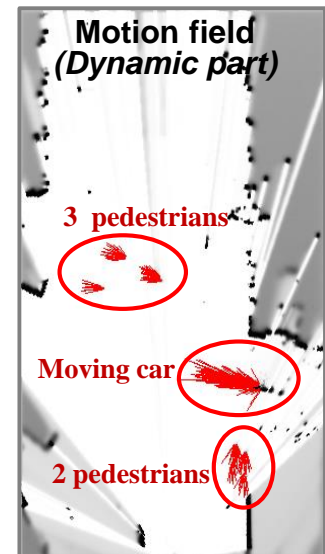
- ❖ **Bayesian Occupancy Filter (BOF)**
 - => Patented by Inria & Probayes
 - => Commercialized by Probayes
 - => Robust to sensing errors & occultation
- ❖ **Used by:** Toyota, Denso, Probayes, Renault, EasyMile, IRT Nanoelec / CEA
- ❖ **Free academic license available**
- ❖ **Industrial licenses: Toyota, EasyMile**

Bayesian Occupancy Filter (BOF) – Main Features

- Estimate **Spatial occupancy** for each cell of the grid $P(O | Z)$
- **Grid update** is performed in each cell in parallel (using *BOF equations*)
- **Extract Motion Field** (using *Bayesian filtering & Fused Sensor data*)
- **Reason at the Grid level** (i.e. *no object segmentation at this reasoning level*)



Sensors data fusion
+
Bayesian Filtering



Exploiting the Dynamic information for a better Understanding of the Scene !!

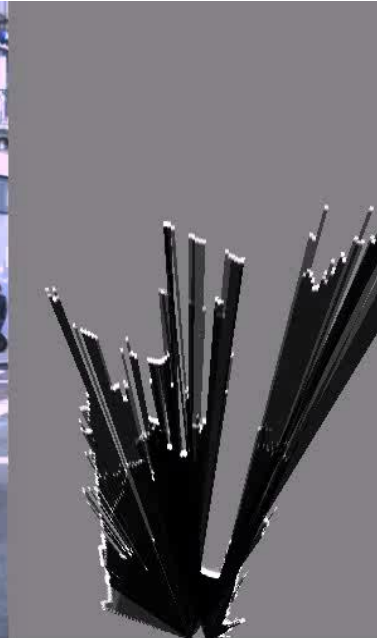
Experimental Results in dense Urban Environments

Observed Urban Traffic scene

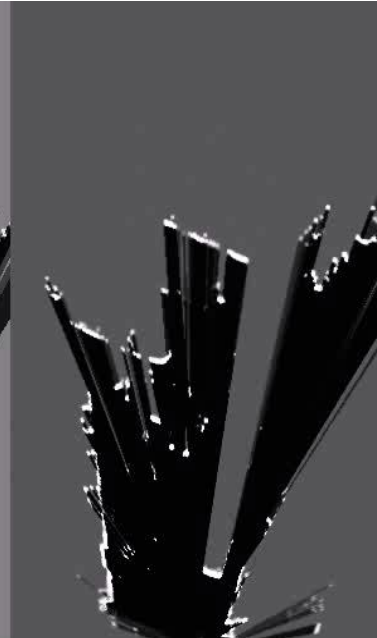


moving vehicle ahead

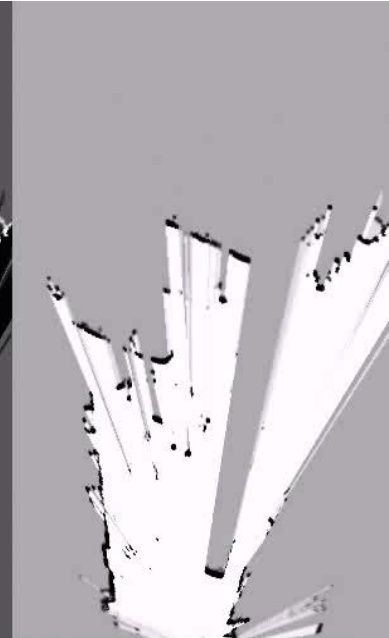
Ego Vehicle (not visible on the video)



OG Left Lidar



OG Right Lidar



OG Fusion
+
Velocity Fields



Bayesian Occupancy Filter – How it works ?

Formalism

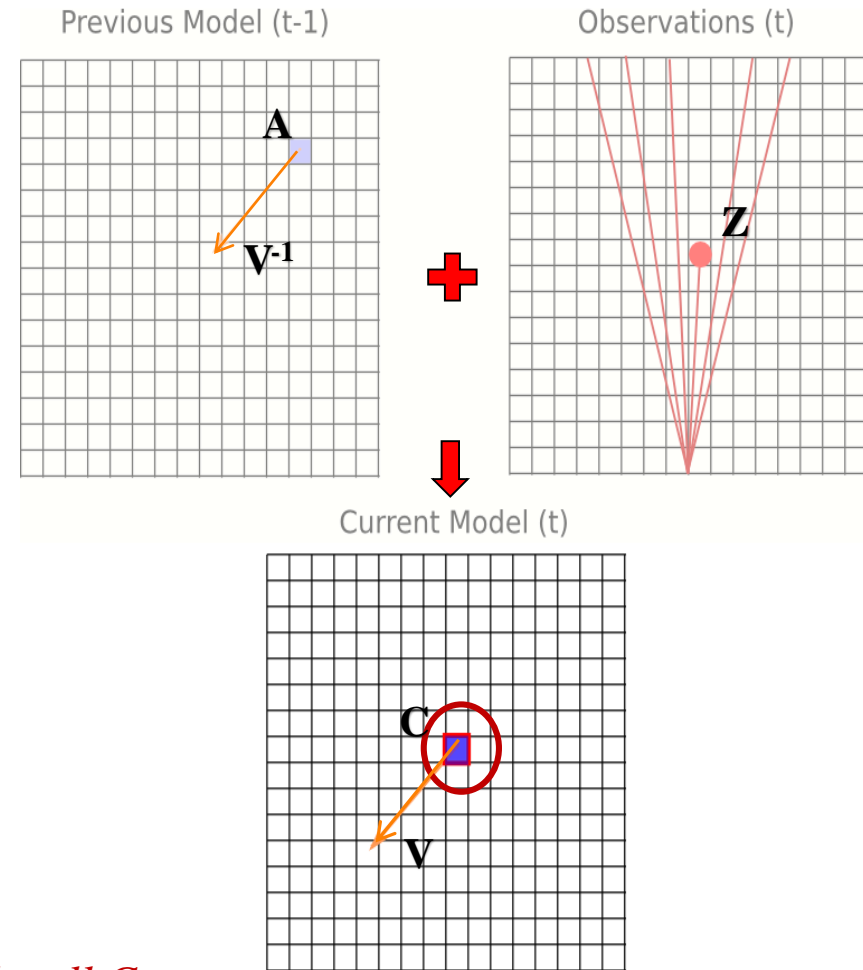
Variables:

- C : current cell
- A : antecedent cell, i.e. the cell from which the occupancy of the current cell comes from
- O : occupancy of the current cell C
- O^{-1} : previous occupancy in the antecedent cell
- V : current velocity
- V^{-1} : previous velocity in the antecedent
- Z : observations (sensor data)

Objective:

Evaluate $P(O \mid V \mid Z \mid C)$

*=> Probability of Occupancy & Velocity for each cell C ,
Knowing the **observations Z** & the **cell location C** in the grid*



Bayesian Occupancy Filter – How it works ?

How to theoretically compute $P(O V | Z C)$?

$$P(O V | Z C) = \lambda \sum_{A O^{-1} V^{-1}} P(C A O O^{-1} V V^{-1} Z)$$

Sum over the possible antecedents **A** and their states (**O⁻¹ V⁻¹**) at time t-1

The joint probability term can be re-written as follows:

$$P(C A O O^{-1} V V^{-1} Z) = P(A) P(O^{-1} V^{-1} | A) P(O V | O^{-1} V^{-1}) P(C | A V) P(Z | O C)$$

Joint probability => used for the update of $P(O V | Z C)$

$P(A)$: Selected as **uniform** (every cell can a priori be an antecedent)

$P(O^{-1} V^{-1} | A)$: Result from the previous iteration

$P(O V | O^{-1} V^{-1})$: **Dynamic model**

$P(C | A V)$: **Indicator function** of the cell **C** corresponding to the “**projection**” in the grid of the antecedent **A** at a given velocity **V**

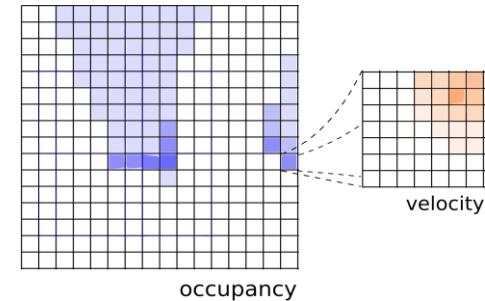
$P(Z | O C)$: **Inverse sensor model**

How to compute $P(OV | Z C)$ in practice?

Initial approach: The classic BOF filtering process

Initial implementation:

- ✓ **Regular grid**
- ✓ **Transition histograms** for every cell (for representing velocities)



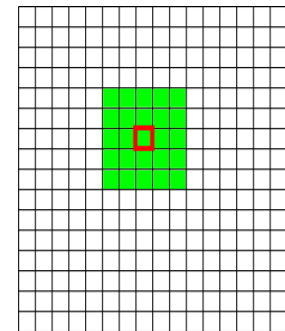
Practical computation:

$$P(OV | ZC) = \lambda \sum_{A O^{-1} V^{-1}} P(C A O O^{-1} V V^{-1} Z)$$

Sum over the possible antecedents A and their states $(O^{-1} V^{-1})$

⇒ Sum over the **neighborhood**, with a **single possible velocity per antecedent A** of equation:

$$P(C A O O^{-1} V V^{-1} Z) = P(A) P(O^{-1} V^{-1} | A) P(O V | O^{-1} V^{-1}) P(C | A V) P(Z | O C)$$



Drawbacks:

- ⇒ Large memory size required (velocity histograms large & almost empty)
- ⇒ Weak accuracy
- ⇒ Temporal & Spatial Aliasing problems

How to compute $P(OV | Z C)$ in practice ?

HSBOF filtering calculation (particles for dynamic component)

$$P(OV | Z C) = \lambda \sum_{A O^{-1} V^{-1}} P(C A O O^{-1} V V^{-1} Z)$$

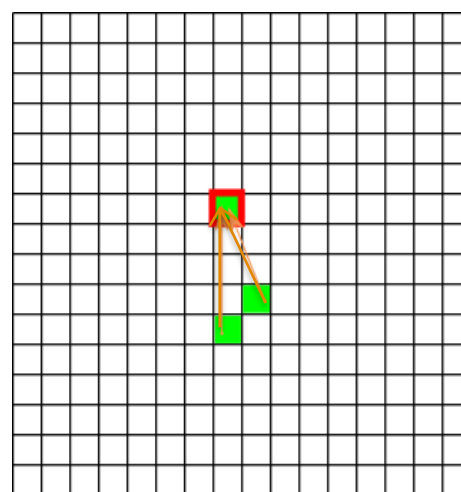
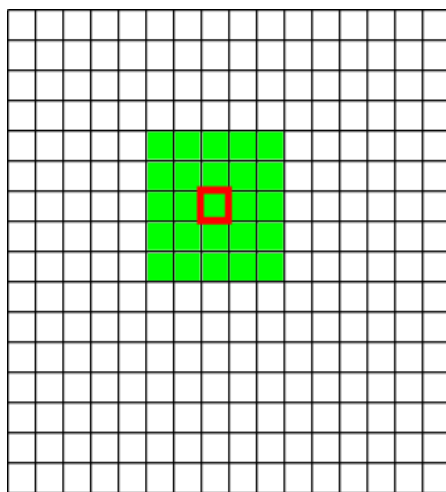
Sum over the neighborhood, with a single velocity per antecedent

A more efficient computation approach :

=> Sum over the particles projected in the cell & their related static parts

$$P(C A O O^{-1} V V^{-1} Z) = P(A) P(O^{-1} V^{-1} | A) P(O V | O^{-1} V^{-1}) \\ P(C | A V) P(Z | O C)$$

Previous
computation approach
(histograms)



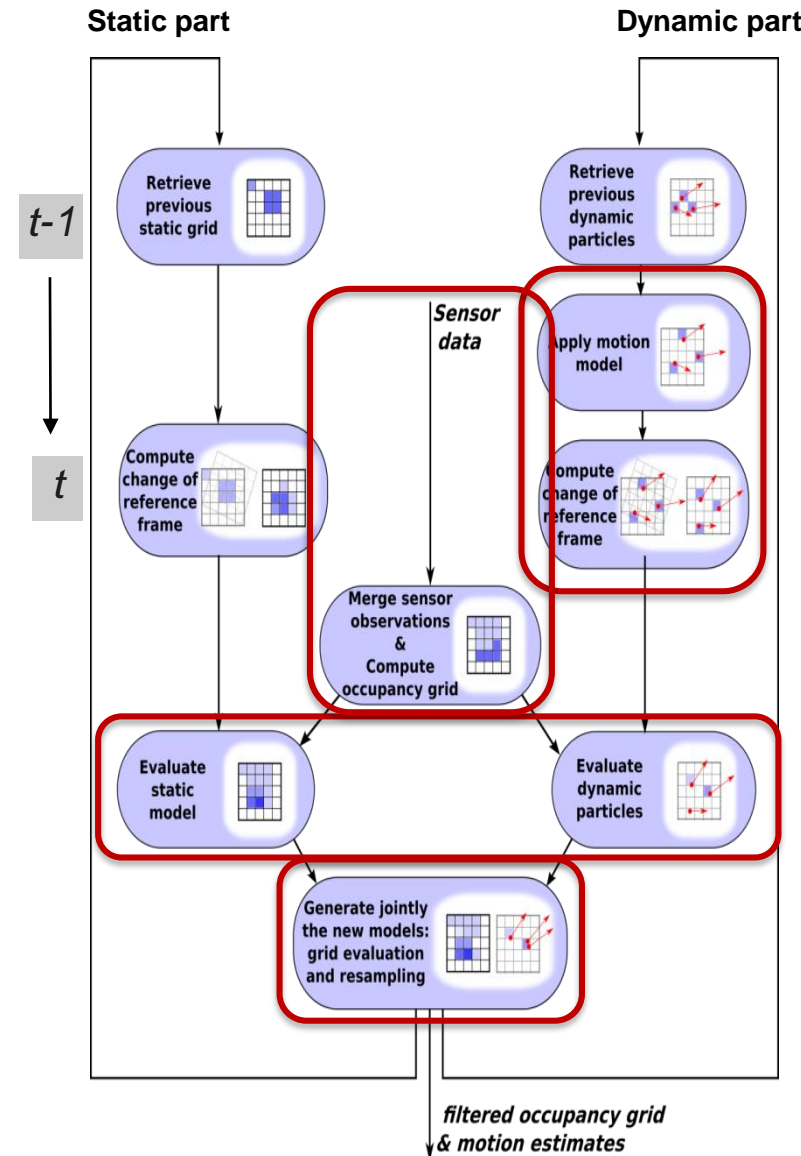
New
computation approach
(particles)

How to compute $P(OV | Z C)$ in practice?

HSBOF updating process (outline of the algorithm)

Main steps in the updating process

- Dynamic part (particles) is “**projected**” in the grid using motion model => *motion prediction*
- Both Dynamic & Static parts are expressed in the **new reference frame** => *moving vehicle frame*
- The two resulting representations are confronted to the **observations** => *estimation step*
- **New representations (static & dynamic)** are jointly evaluated and particles re-sampled

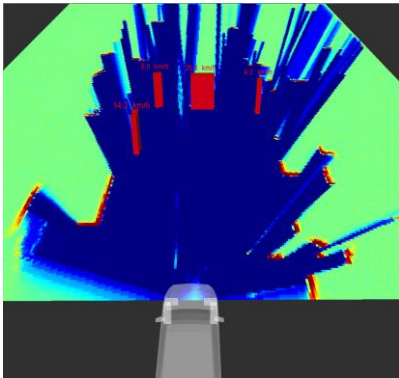


Recent implementations & Improvements

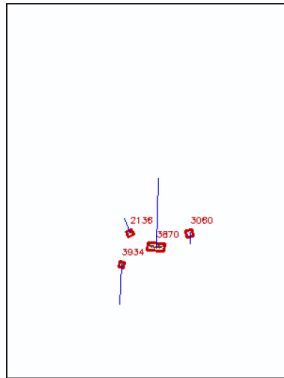


*Several implementations (models & algorithms) more and more adapted to **Embedded constraints & Scene complexity***

- ❖ Hybrid Sampling Bayesian Occupancy Filter (HSBOF, 2014) [Negre et al 14] [Rummelhard et al 14]
=> **Drastic memory size reduction** (factor 100) + **Increased efficiency** (complex scenes)
+ **More accurate Velocity estimation** (using Particles & Motion data from ego-vehicle)
- ❖ Conditional Monte-Carlo Dense Occupancy Tracker (CMCDOT, 2015) [Rummelhard et al 15]
=> **Increased efficiency using “state data”** (Static, Dynamic, Empty, **Unknown**) + **Integration of a “Dense Occupancy Tracker”** (Object level, Using particles propagation & ID)
- ❖ CMCDOT + Ground Estimator (Patent 2017) [Rummelhard et al 17]
=> **Ground shape estimation & Improve obstacle detection** (avoid false detections on the ground)



Grid & Pseudo-objects



Tracked Objects

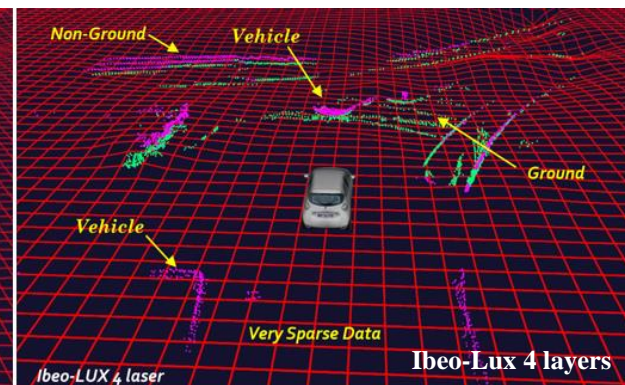
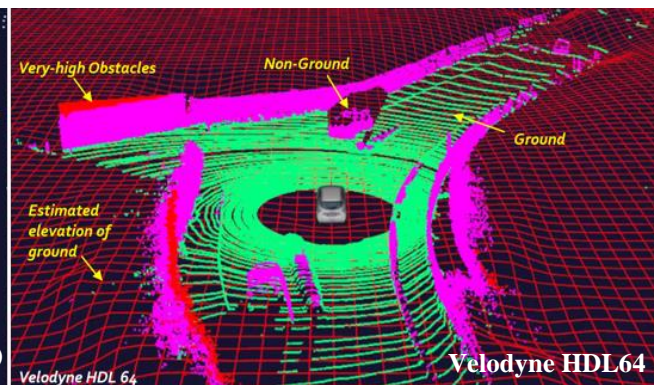
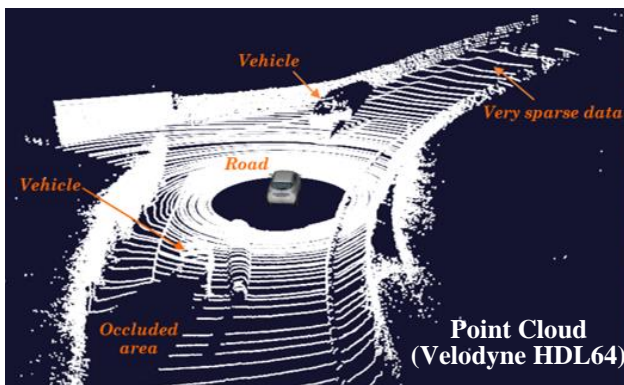


Classification (using Deep Learning)

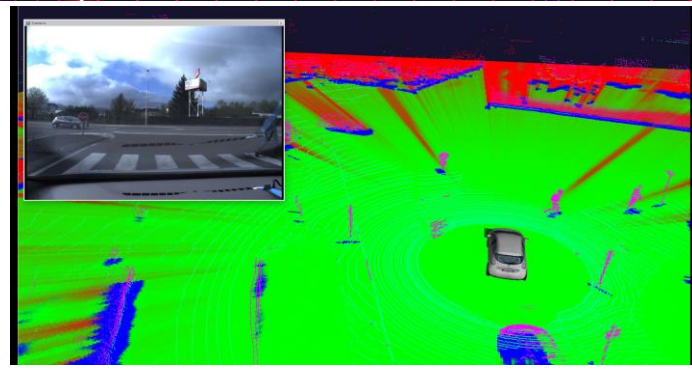
Detection & Tracking
& Classification

Ground Estimation & Point Cloud Classification

- ⇒ Smart OG generation taking into account the ground shape & height of laser impacts
- ⇒ Process properly sensors data (for OG & DATMO) & Avoid false detections on the road surface



- **Ground estimation** : 1m x 1m ground node resolution
⇒ ground-points in green, obstacles in pink / red
- **Occupancy grid** : Images 0.1m x 0.1m occupancy grid resolution
⇒ green for free space, blue occupied, red unknown



- ❖ Ground model constructed using a *Spatio-Temporal Conditional Random Field*, estimated through efficient parallelized process [1]
- ❖ Model accurate enough to represent rolling roads & Efficient enough for **real-time** performances on embedded devices
- ❖ The complete system (including CMCDOT) has been implemented on a Nvidia Tegra X1



[1] Ground estimation and point cloud segmentation using spatio-temporal conditional random field, Rummelhard et al, IV 2017, Redondo Beach, June 2017

Integration on a commercial vehicle



- **POC 2017: Complete system implemented on Nvidia TX1**, and easily connected to the shuttle system network *in a few days* (using ROS)



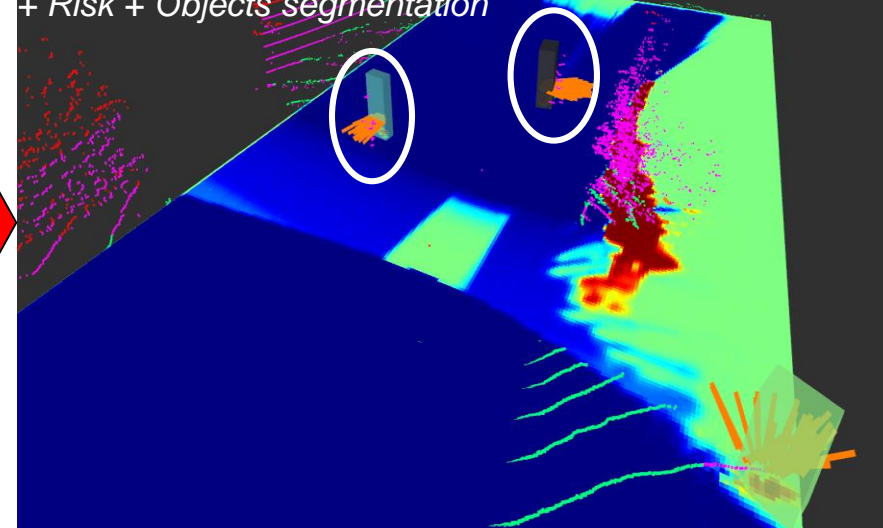
- **Shuttle sensor data** has been fused and processed in **real-time**, with a successful Detection & Characterization of the **Moving & Static Obstacles**
- **Full integration on a commercial product** under development with an industrial company (confidential)

Point cloud classification, with a pedestrian behind the shuttle, and a car in front

Detected moving objects

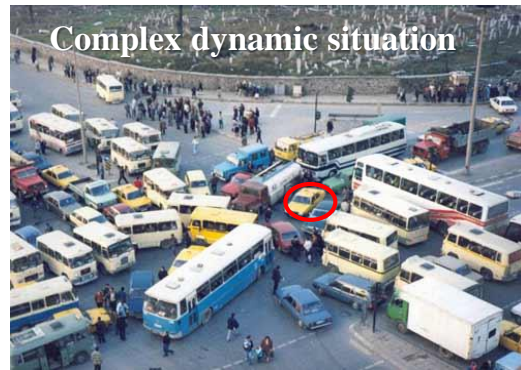
2 Velodyne VLP16
+
4 LMS mono-layer

CMCDOT filtered Occupancy Grid + Inferred Velocities
+ Risk + Objects segmentation



Key Technology 2: Risk Assessment & Decision

=> Decision-making for avoiding Pending & Future Collisions



□ Main challenges

Uncertainty, Partial Knowledge, World changes, Human in the loop + Real time

□ Approach: Prediction + Risk Assessment + Bayesian Decision-making

- ✓ Reason about *Uncertainty & Contextual Knowledge* (using *History & Prediction*)
- ✓ Estimate probabilistic Collision Risk at a given *time horizon* $t+\delta$
- ✓ Make Driving Decisions by taking into account the *Predicted behavior* of all the observed surrounding traffic participants (cars, cycles, pedestrians ...) & *Social / Traffic rules*

Underlying Conservative Prediction Capability

⇒ *Application to Conservative Collision Anticipation*

[Coué & Laugier IJRR 05]

Autonomous
Vehicle (Cycab)



Parked Vehicle
(occultation)

**Pioneer
Results
(2005)**

Thanks to the prediction capability of the BOF technology, the Autonomous Vehicle “anticipates” the behavior of the pedestrian and brakes *(even if the pedestrian is temporarily hidden by the parked vehicle)*

Step 1: Short-term collision risk – *Main features*

=> *Grid level & Conservative motion hypotheses (proximity perception)*

□ Main Features

- Detect “*Risky Situations*” a few seconds ahead (3-5s)
- Risky situations are *both localized in Space & Time*
 - ⇒ *Conservative Motion Prediction in the grid (Particles & Occupancy)*
 - ⇒ *Collision checking with Car model (shape & velocity) for every future time steps (horizon h)*
- Resulting information can be used for choosing *Avoidance Maneuvers*

Proximity perception: $d < 100m$ and $t < 5s$

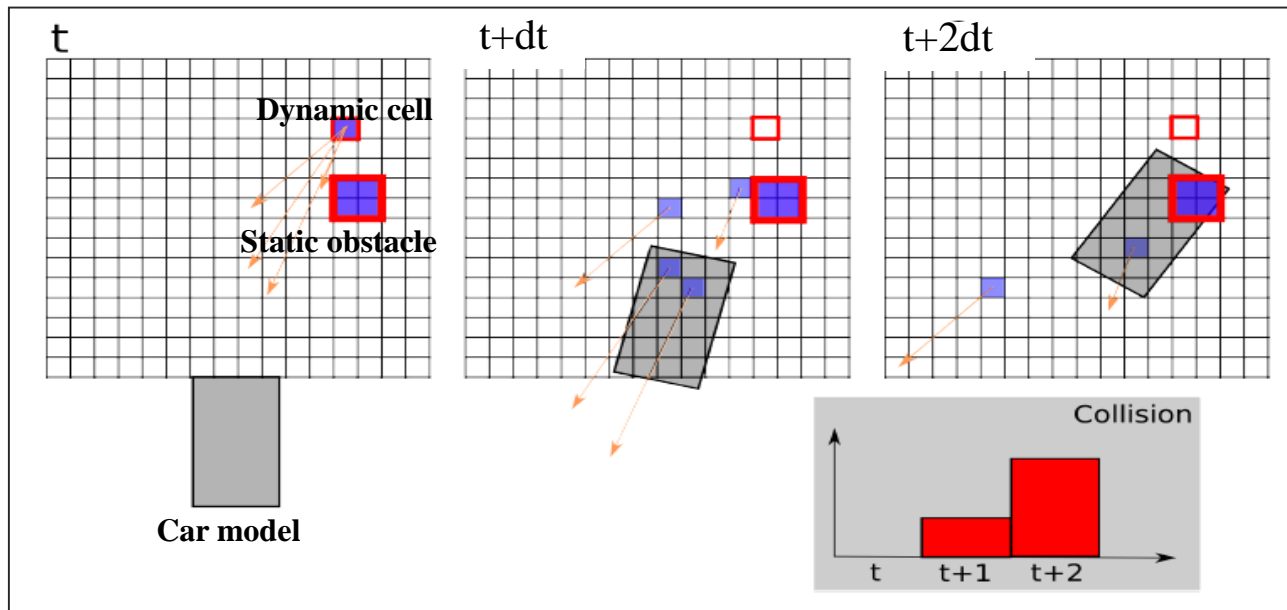
$\delta = 0.5s$ => *Precrash*

$\delta = 1s$ => *Collision mitigation*

$\delta > 1.5s$ => *Warning / Emergency Braking*

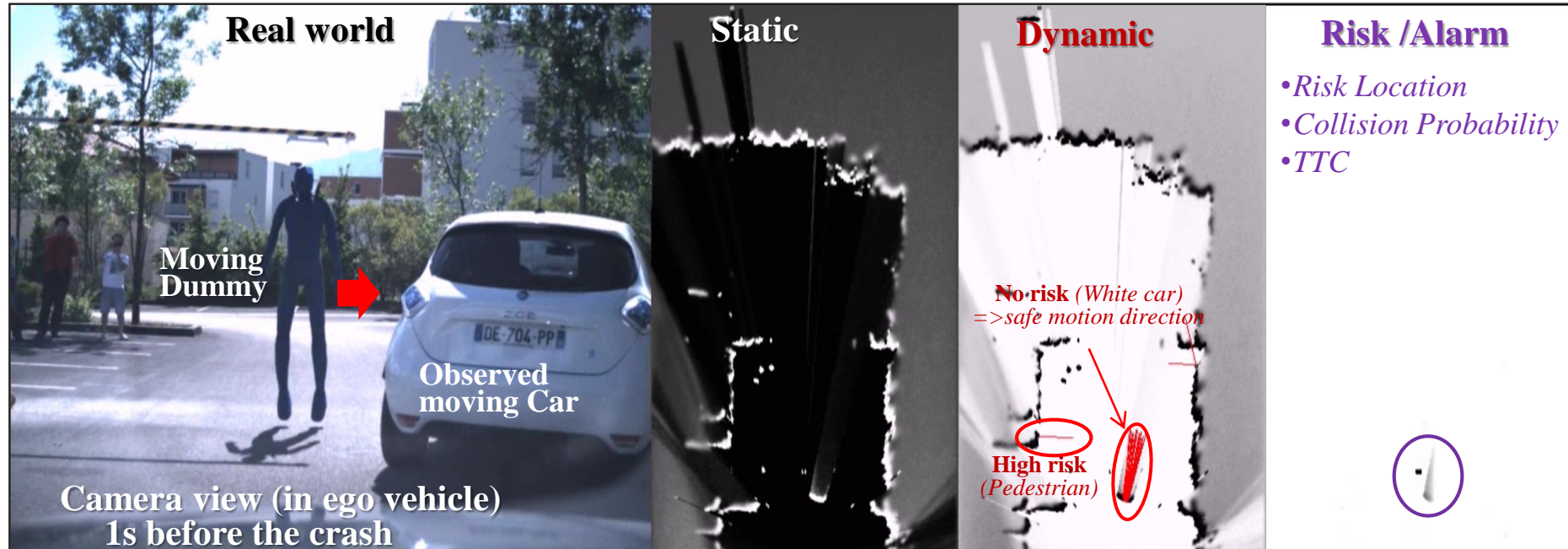
□ Collision Risk Estimation: *Integration of risk over a time range $[t, t + \delta]$*

=> *Projecting over time the estimated Scene changes (DP-Grid) & Car Model (Shape + Motion)*



Short-term collision risk – System outputs (real-time)

=> *Static & Dynamic grids + Risk assessment*



- **FAQ :** What happen if some velocities change after that the collision risk for the next 3s has been evaluated ?
- **Answer:** The collision risk is recomputed at the next time step (i.e max 40ms after the change of dynamics).

Short-term collision risk – *Experimental results*

- ⇒ Detect potential upcoming collisions
- ⇒ Reduce drastically false alarms



Crash test with no automatic braking

CMCDOT – Complete process illustration (video)



Sensor data

Inria
informatics mathematics

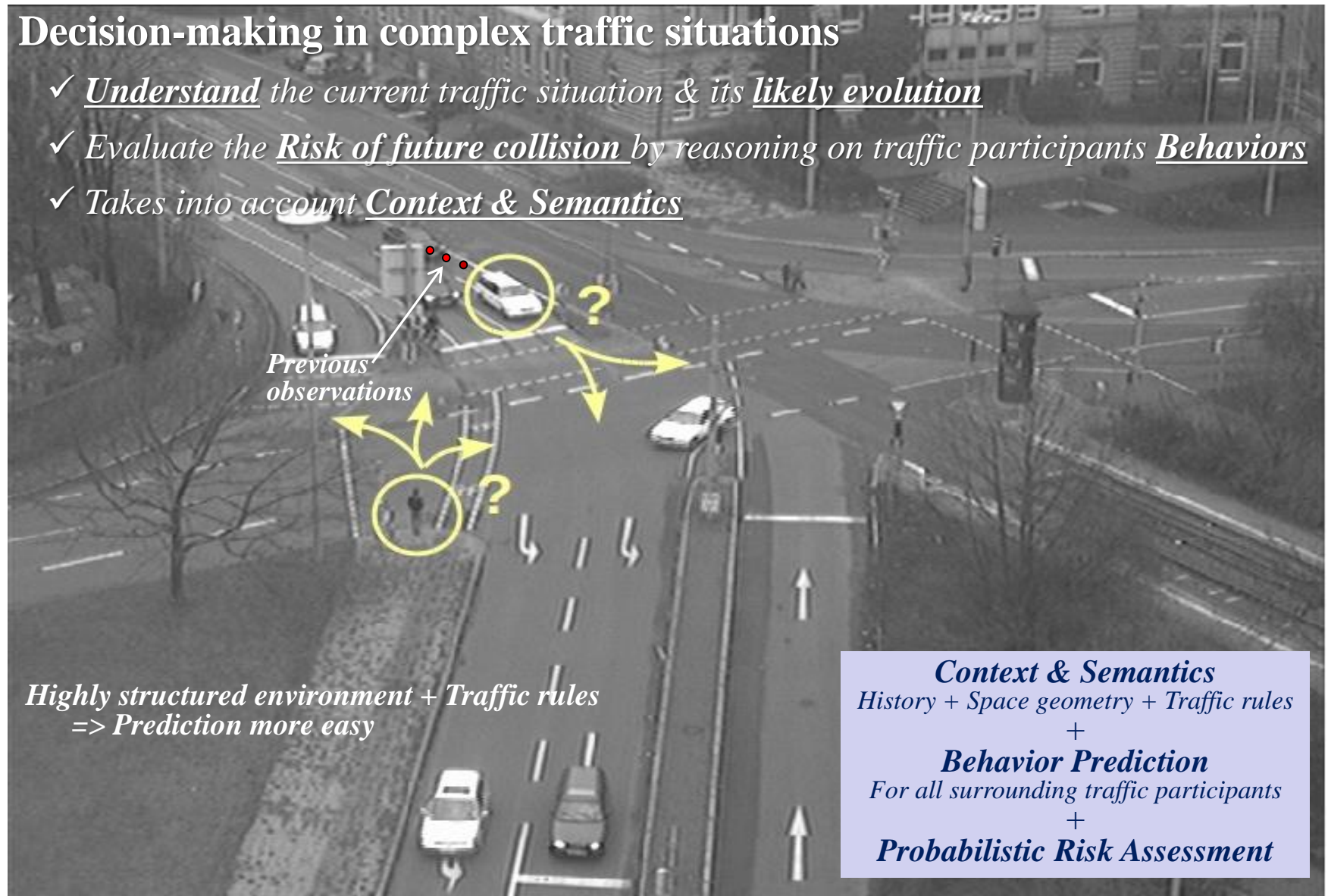


Step 2: Generalized Risk Assessment (Object level)

- ⇒ Increasing time horizon & complexity using context & semantics
- ⇒ Key concept: **Behaviors Modeling & Prediction**

Decision-making in complex traffic situations

- ✓ Understand the current traffic situation & its likely evolution
- ✓ Evaluate the Risk of future collision by reasoning on traffic participants Behaviors
- ✓ Takes into account Context & Semantics



*Highly structured environment + Traffic rules
=> Prediction more easy*

Context & Semantics
History + Space geometry + Traffic rules
+
Behavior Prediction
For all surrounding traffic participants
+
Probabilistic Risk Assessment

Behavior-based Collision risk (Object level)

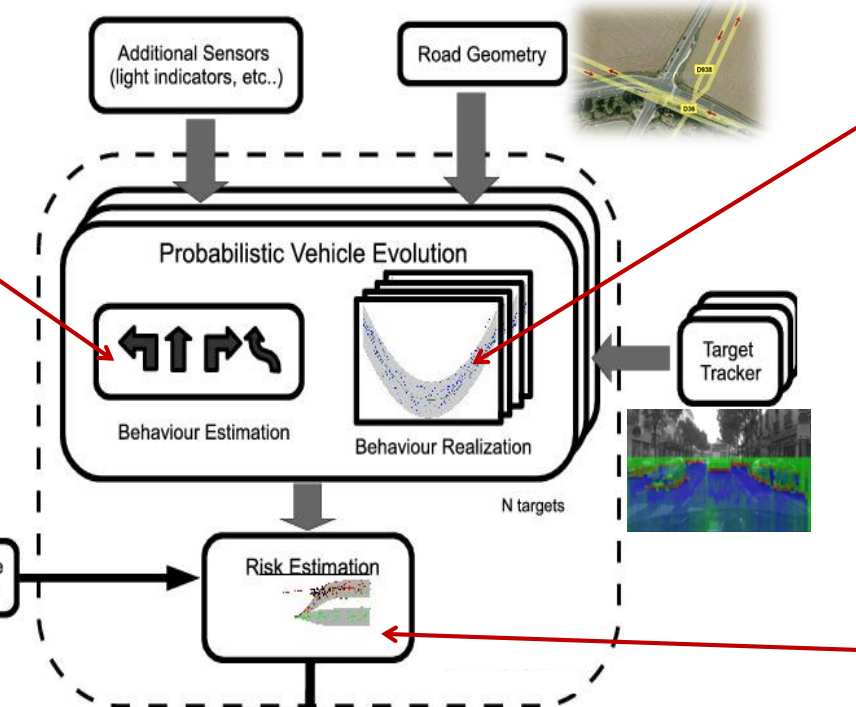
=> Increasing time horizon & complexity + Reasoning on Behaviors

Approach 1: Trajectory prediction & Collision Risk Assessment

Patent Inria & Toyota & Probayes 2010 + [Tay thesis 09] [Laugier et al 11]

Behavior modeling & learning
+
Behavior Prediction

Layered HMM

$$P(B_t|O_{1:t}) = L_{B_t}(O_{1:t}) \sum_{B_{t-1}} P(B_{t-1})P(B_t|B_{t-1})$$


From behaviors to trajectories

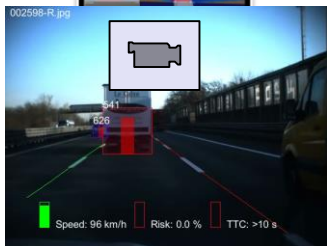
Gaussian Process + LSCM

Collision risk assessment (Probabilistic)

MC simulation



Courtesy Probayes



Experimental Results
Behavior prediction & Risk Assessment on highway
Probayes & Inria & Toyota

Behavior-based Collision risk (*Object level*)

=> *Increasing time horizon & complexity + Reasoning on Behaviors*

Approach 2: Intention & Expectation comparison

=> *Complex scenarios with Interdependent Behaviors & Human Drivers*



[Lefevre thesis 13] [Lefevre & Laugier IV'12, Best student paper]

Patent Inria & Renault 2012 (*risk assessment at road intersection*)

Patent Inria & Berkeley 2013 (*postponing decisions for safer results*)



A Human-like reasoning paradigm => *Detect Drivers Errors & Colliding behaviors*

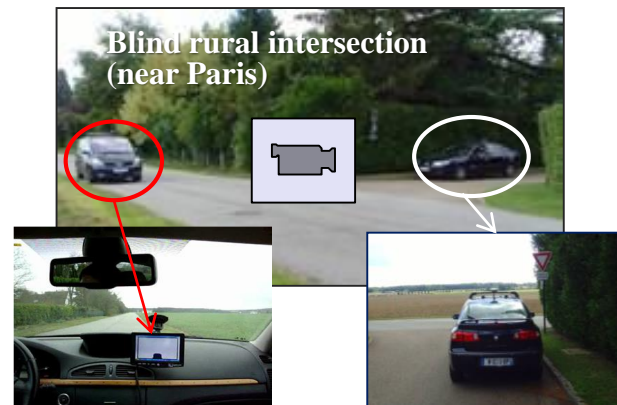
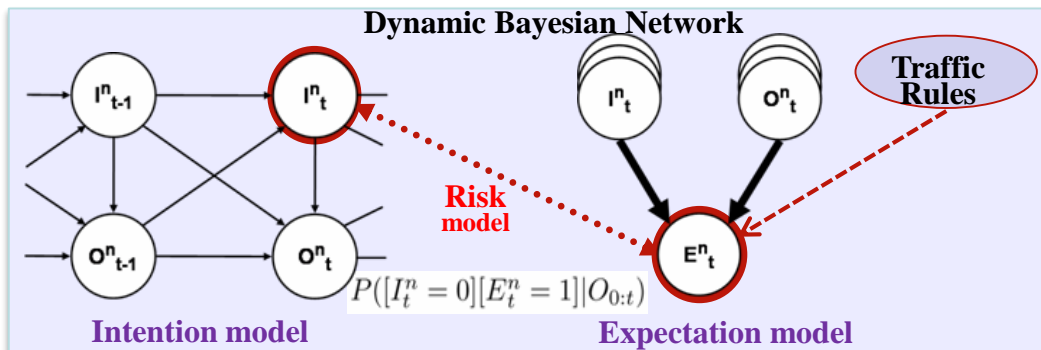
✓ *Estimating "Drivers Intentions" from Vehicles States Observations ($X Y \theta S TS$) => Perception or V2V*

✓ *Inferring "Behaviors Expectations" from Drivers Intentions & Traffic rules*

✓ *Risk = Comparing Maneuvers Intention & Expectation*

=> *Taking traffic context into account (Topology, Geometry, Priority rules, Vehicles states)*

=> *Digital map obtained using "Open Street Map"*



Experimental Vehicles & Connected Perception Units

Toyota Lexus



ROS



Renault Zoé
Velodyne 3D lidar



Connected Perception Unit

=> Same embedded perception systems than in vehicles

Nvidia GTX Titan X
Generation Maxwell



Nvidia GTX Jetson TK1
Generation Maxwell



Nvidia GTX Jetson TX1
Generation Maxwell



Software / Hardware Integration – GPU implementation



- Highly parallelizable framework, **27 kernels** over cells and particles
=> Occupancy, speed estimation, re-sampling, sorting, prediction
- Real-time implementation (20 Hz), optimized using Nvidia profiling tools

Results:

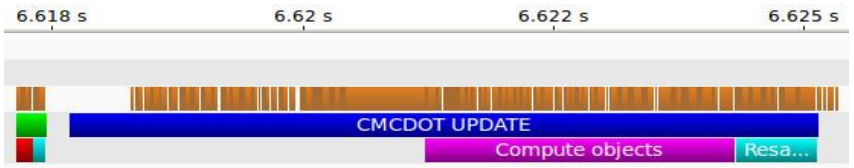
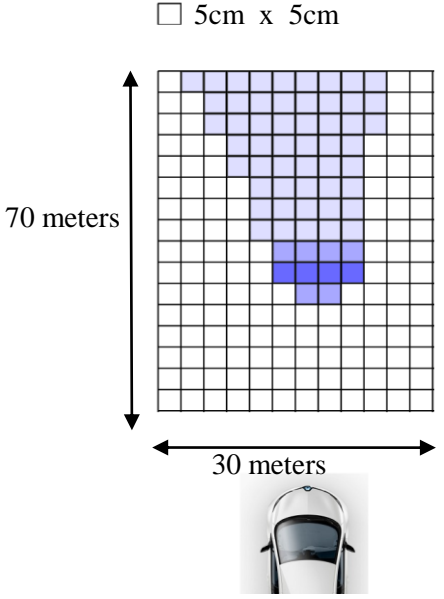
- Configuration with 8 Lidar layers (2x4)
- Grid: 1400 x 600 (840 000 cells) + Velocity samples: 65 536



=> Jetson TK1: *Grid Fusion 17ms, CMCDOT 70ms*



=> Jetson TX1: Grid Fusion 0.7ms, CMCDOT 17ms

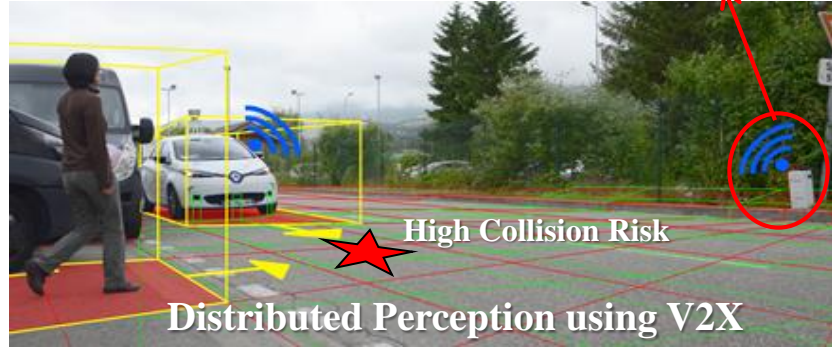
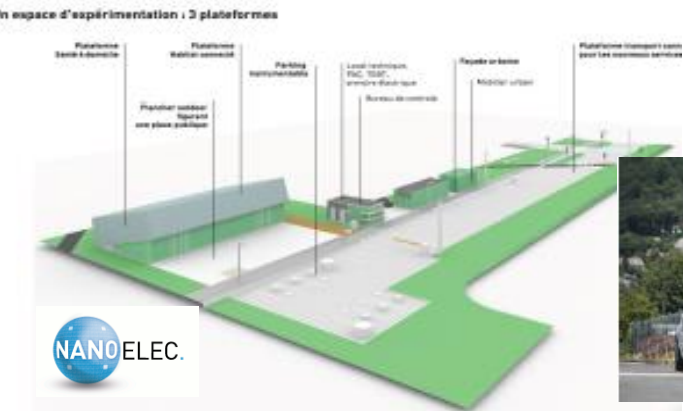


Experimental Areas



Connected Perception Unit

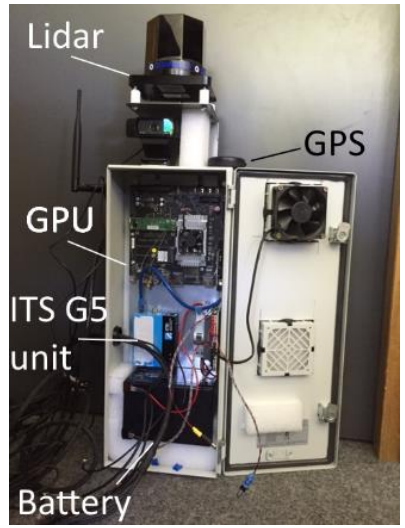
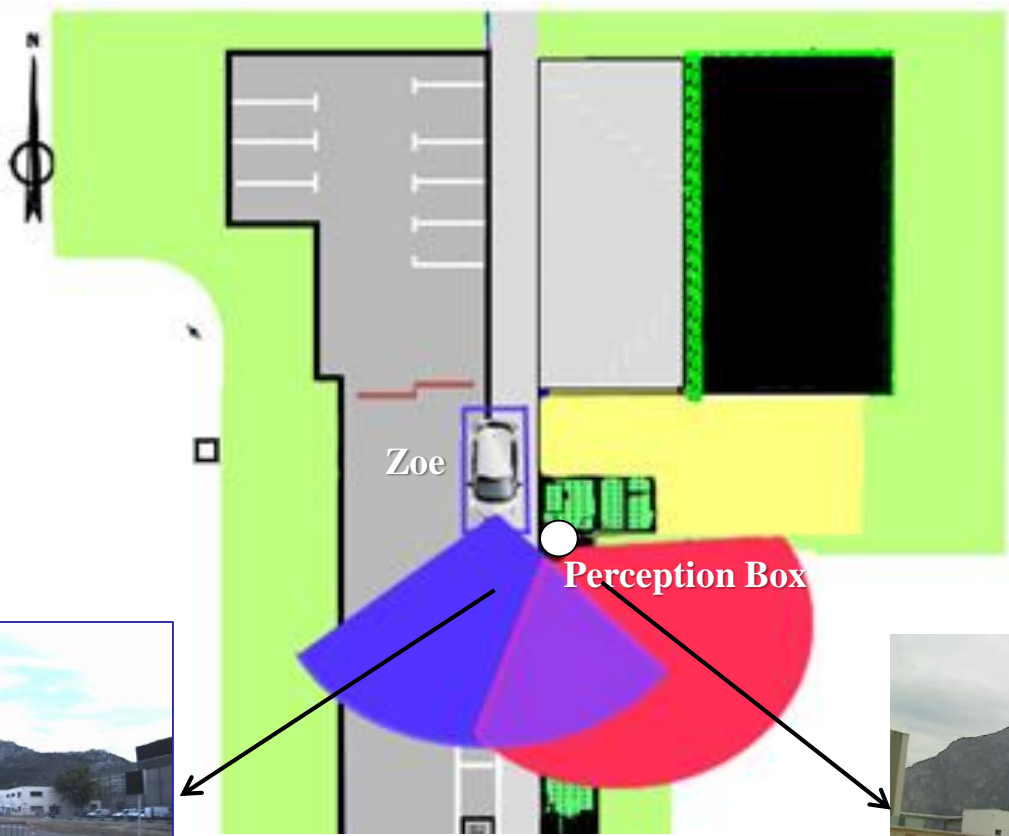
Protected experimental area



Open real traffic (Urban & Highway)



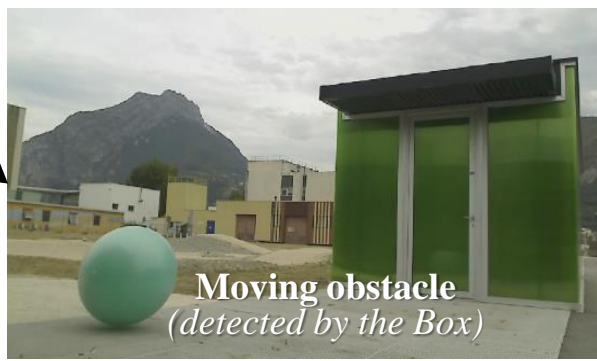
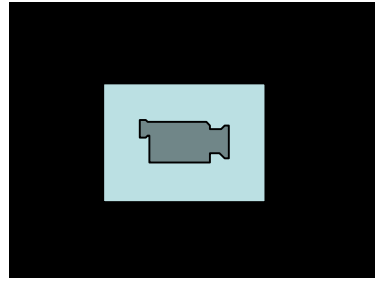
V2X: Distributed Perception experiment using CMCDOT



Connected Perception Unit



Camera Image provided by the Zoe vehicle



Moving obstacle (detected by the Box)

Camera Image provided by the Perception box

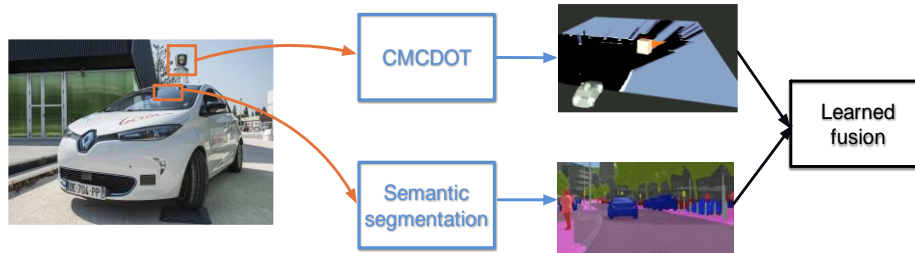
Current & Future work

□ New models/algos for integration in various platforms & dynamics

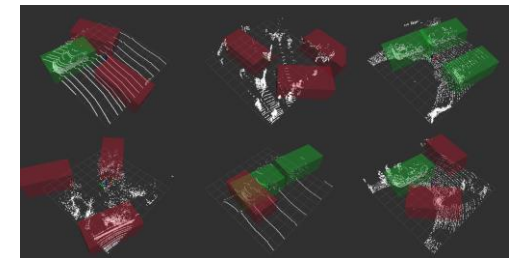


Integration within the Control Units of EasyMile EZ10 shuttle & Iveco bus & Renault Zoe

□ News approaches & models for constructing “semantic grids” using CNN

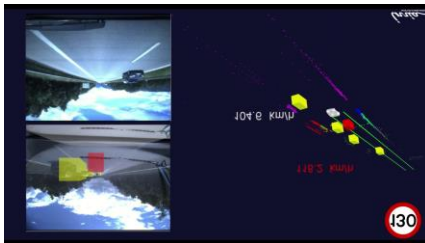


Fusing CMCDOT output (OG) with semantic output from RGB cameras (coop Toyota)
=> *Patent application & Publications (IROS 2018 & ICARCV 2018)*

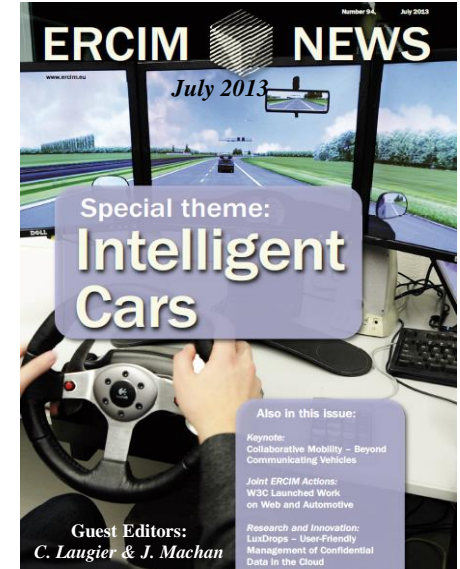
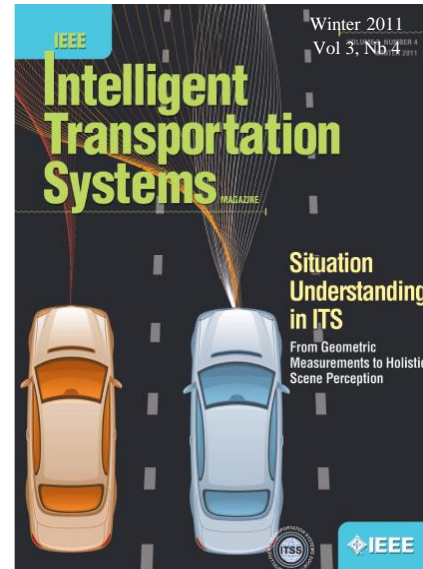
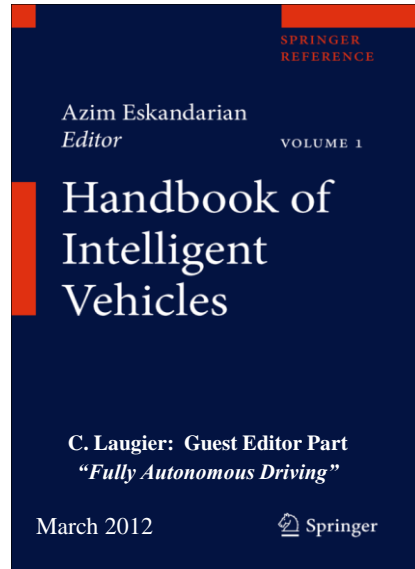


Semantic segmentation in a 3D point cloud
=> *Master Thesis*

□ Driving behavior learning & prediction for autonomous driving



- Driver behavior modeling using Inverse Reinforcement Learning
 - Combining Model-based prediction with Dynamic evidence to estimate lane change intentions
- => *2 Patents application & publications (ITSC 2016, ICRA 2017, ICRA 2018)*



Thank You Any questions ?

