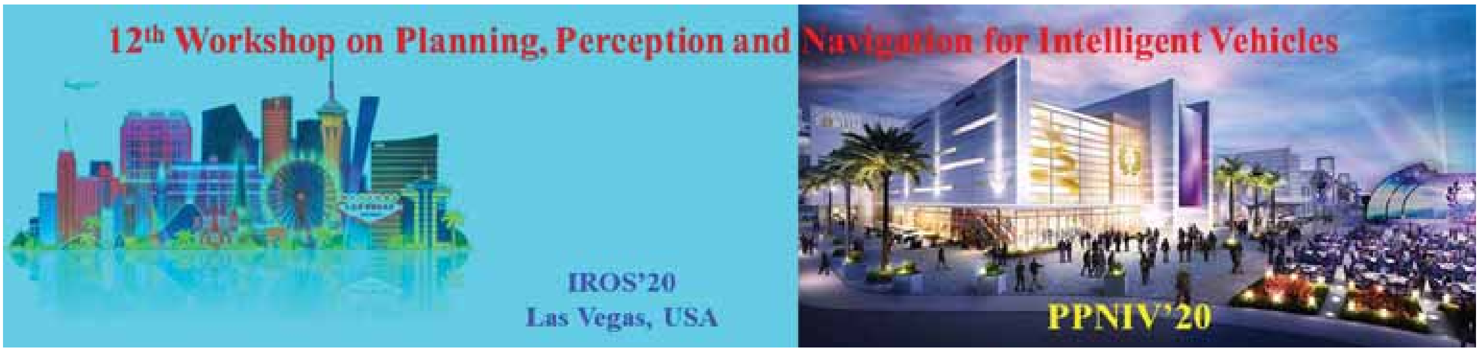

12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS20

12th International workshop on

Planning, Perception and Navigation for Intelligent Vehicles

Full Day Workshop

October 25th, 2020, Las Vegas, USA

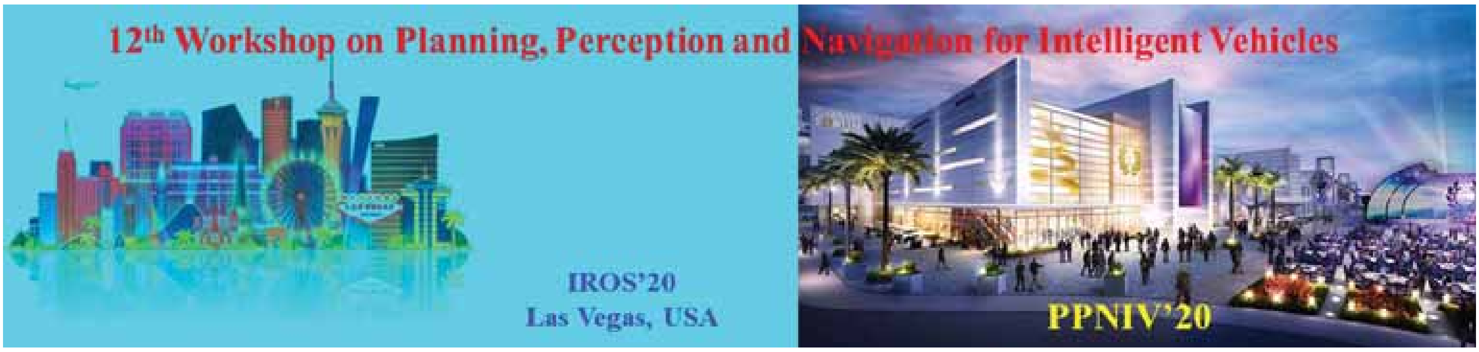
https://project.inria.fr/ppniv20/

Organizers

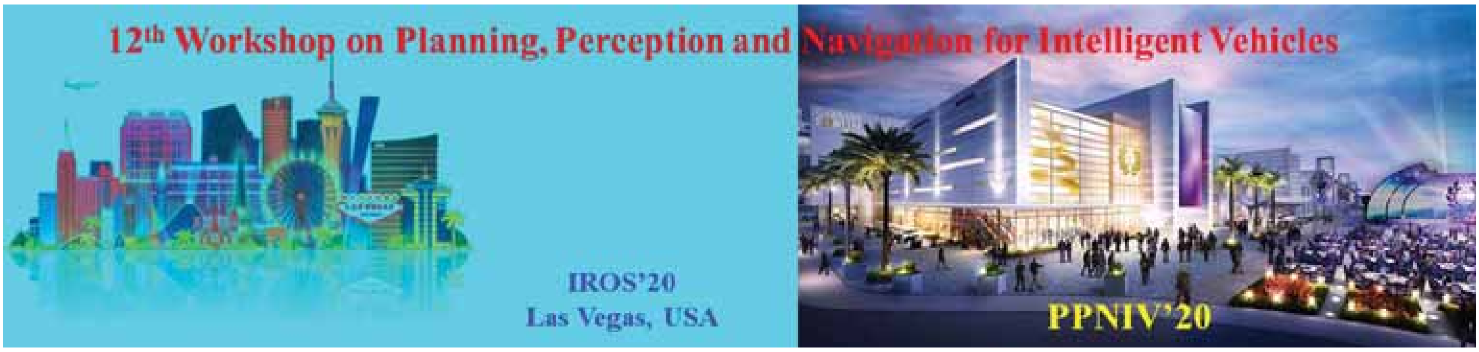
Pr Marcelo Ang (NUS, Singapore)
DR Christian Laugier (INRIA, France),
Pr Philippe Martinet (INRIA, France),
Pr Denis Wolf (University Sao Paulo, Brazil)

Contact

Director of Research Philippe Martinet
Inria - CHORALE team
2004 route des Lucioles, 06902 Sophia-Antipolis, FRANCE
Phone: +33 240 376 975, Sec : +33 240 376 934, Fax : +33 240 376 6930
Email: Philippe.Martinet@inria.fr
Home page: http://www-sop.inria.fr/members/Philippe.Martinet/


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Foreword

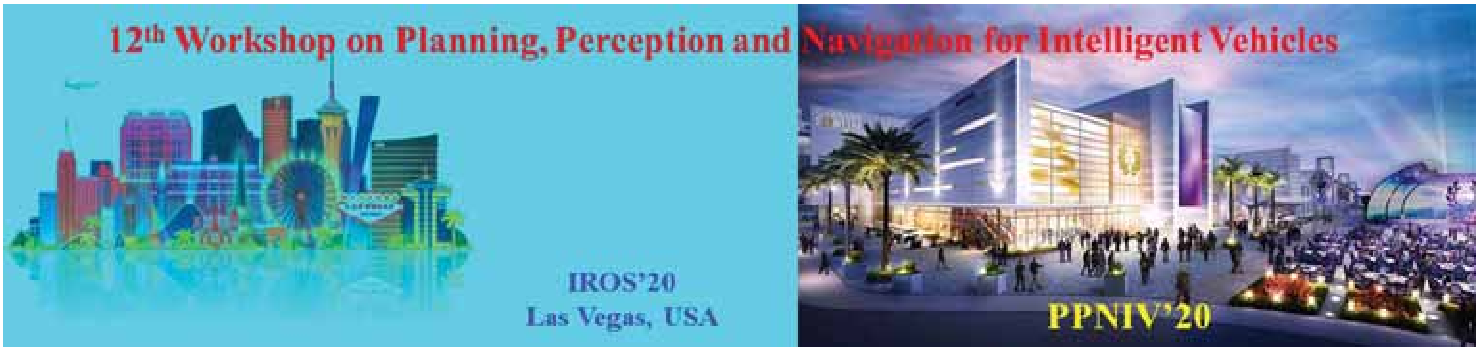
The purpose of this workshop is to discuss topics related to the challenging problems of autonomous navigation and of driving assistance in open and dynamic environments. Technologies related to application fields such as unmanned outdoor vehicles or intelligent road vehicles will be considered from both the theoretical and technological point of views. Several research questions located on the cutting edge of the state of the art will be addressed. Among the many application areas that robotics is addressing, transportation of people and goods seem to be a domain that will dramatically benefit from intelligent automation. Fully automatic driving is emerging as the approach to dramatically improve efficiency while at the same time leading to the goal of zero fatalities. This workshop will address robotics technologies, which are at the very core of this major shift in the automobile paradigm. Technologies related to this area, such as autonomous outdoor vehicles, achievements, challenges and open questions would be presented. Main topics include: Road scene understanding, Lane detection and lane keeping, Pedestrian and vehicle detection, Detection, tracking and classification, Feature extraction and feature selection, Cooperative techniques, Collision prediction and avoidance, Advanced driver assistance systems, Environment perception, vehicle localization and autonomous navigation, Real-time perception and sensor fusion, SLAM in dynamic environments, Mapping and maps for navigation, Real-time motion planning in dynamic environments, Human-Robot Interaction, Behavior modeling and learning, Robust sensor-based 3D reconstruction, Modeling and Control of mobile robot, Deep learning applied in autonomous driving, Deep reinforcement learning applied in intelligent vehicles.

Previously, several workshops were organized in the near same field. The 1st edition PPNIV'07 of this workshop was held in Roma during ICRA'07 (around 60 attendees), the second PPNIV'08 was in Nice during IROS'08 (more than 90 registered people), the third PPNIV'09 was in Saint-Louis (around 70 attendees) during IROS'09, the fourth edition PPNIV'12 was in Vilamoura (over 95 attendees) during IROS'12, the fifth edition PPNIV'13 was in Vilamoura (over 135 attendees) during IROS'13, the sixth edition PPNIV'14 was in Chicago (over 100 attendees) during IROS14, the seventh edition PPNIV'15 was in Hamburg (over 150 attendees) during IROS15, the heigth edition PPNIV'16 was in Rio de Janeiro (over 100 attendees) during ITSC16, the nineth edition PPNIV17 was in Vancouver during IROS17 (over 170 attendees), the 10th edition PPNIV'18 was in Madrid during IROS18 (over 350 attendees), and the 11th edition PPNIV'19 has gathered over 300 attendees in Macau. For the first time, PPNIV20 will be organized as a virtual event due to the sanitary conditions in relation to COVID19.

In parallel, we have also organized SNODE'07 in San Diego during IROS'07 (around 80 attendees), MEPPC08 in Nice during IROS'08 (more than 60 registered people), SNODE'09 in Kobe during ICRA'09 (around 70 attendees), RITS'10 in Anchrorage during ICRA'10 (around 35 attendees), PNAVHE11 in San Francisco during the last IROS11 (around 50 attendees), and the last one WMEPC14 in Hong Kong during the last ICRA14 (around 65 attendees),

This workshop is composed with 4 invited talks and 10 selected papers.

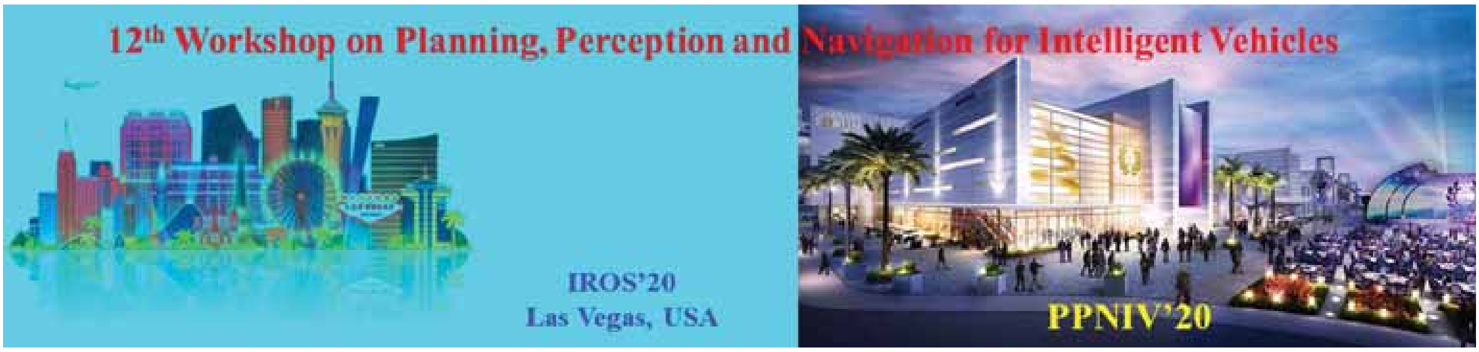
Intended Audience concerns researchers and PhD students interested in mobile robotics, motion and action planning, robust perception, sensor fusion, SLAM, autonomous vehicles, human-robot interaction, and intelligent transportation systems. Some peoples from the mobile robot industry and car industry are also welcome.


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

This workshop is made in relation with IEEE RAS: RAS Technical Committee on "Autonomous Ground Vehicles and Intelligent Transportation Systems" (http://tab.ieee-ras.org/).

Christian Laugier, Philippe Martinet, Marcelo Ang and Denis Wolf


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

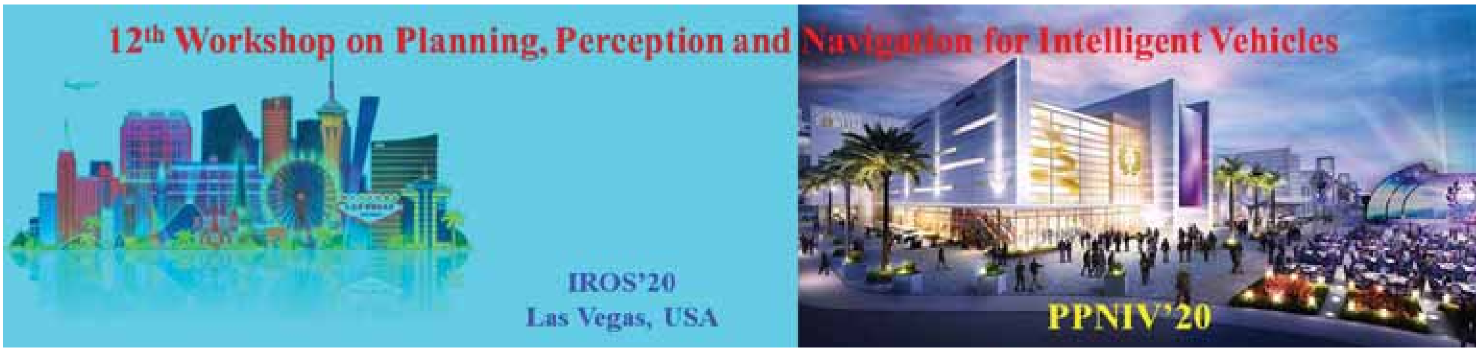
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Keynote speaker: **Wolfram Burgard**
(University of Frieburg, Germany)

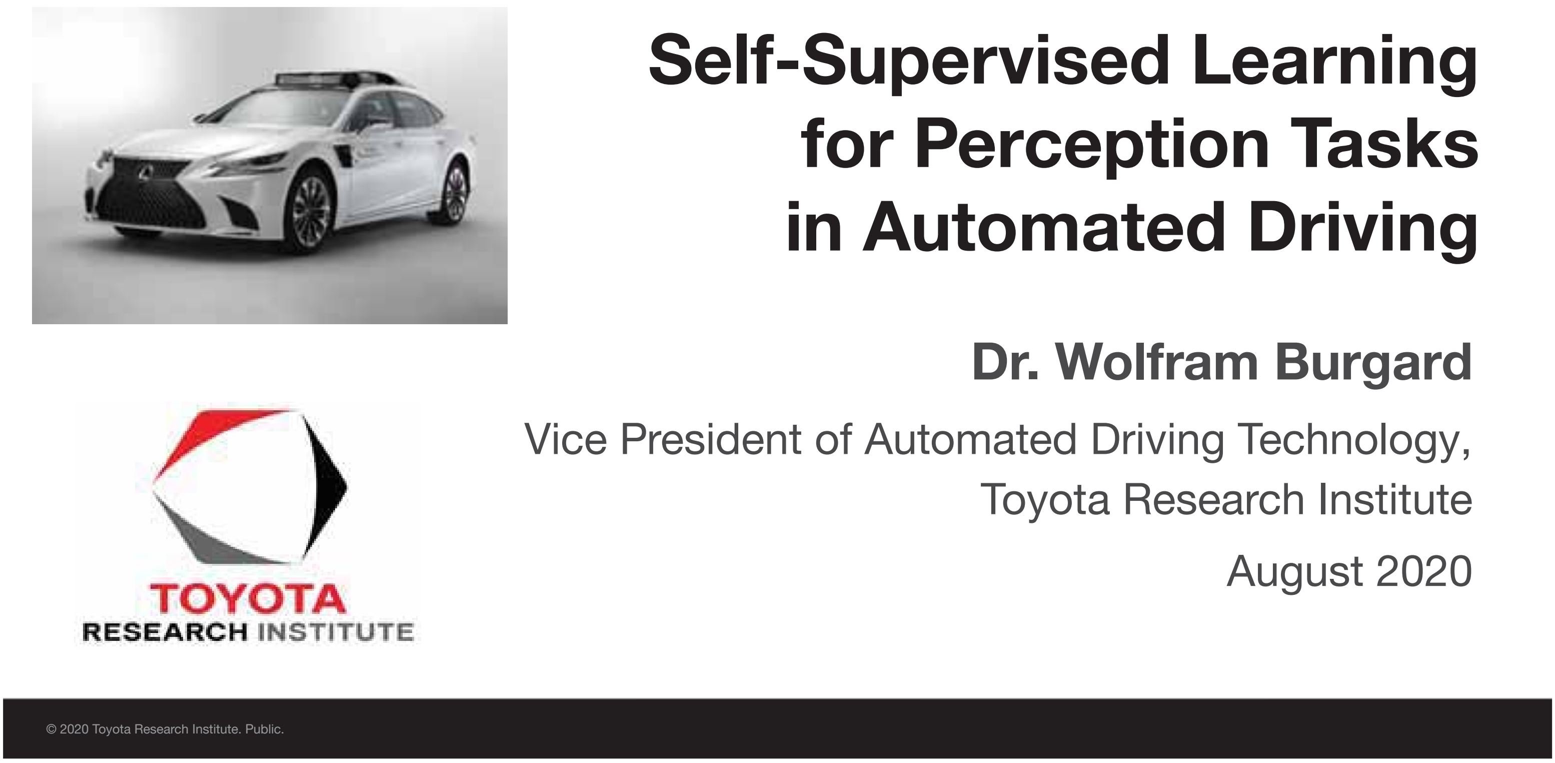
Self-Supervised Learning for Perception Tasks in Automated Driving

Abstract: At the Toyota Research Institute we are following the one-system-two-modes approach to building truly automated cars. More precisely, we simultaneously aim for the L4/L5 chauffeur application and the the guardian system, which can be considered as a highly advanced driver assistance system of the future that prevents the driver from making any mistakes. TRI aims to equip more and more consumer vehicles with guardian technology and in this way to turn the entire Toyota fleet into a giant data collection system. To leverage the resulting data advantage, TRI performs substantial research in machine learning and, in addition to supervised methods, particularly focuses on unsupervised and self-supervised approaches. In this presentation, I will present three recent results regarding self-supervised methods for perception problems in the context of automated driving. I will present novel approaches to inferring depth from monocular images and a new approach to panoptic segmentation.

Biograpghy: Wolfram Burgard is VP for Automated Driving Technology at the Toyota Research Institute. He is on leave from his professorship at the University of Freiburg where he heads the research group for Autonomous Intelligent Systems. Wolfram Burgard is known for his contributions to mobile robot navigation, localization and SLAM (simultaneous localization and mapping). He has published more than 350 papers in the overlapping area of robotics and artificial intelligence.


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems



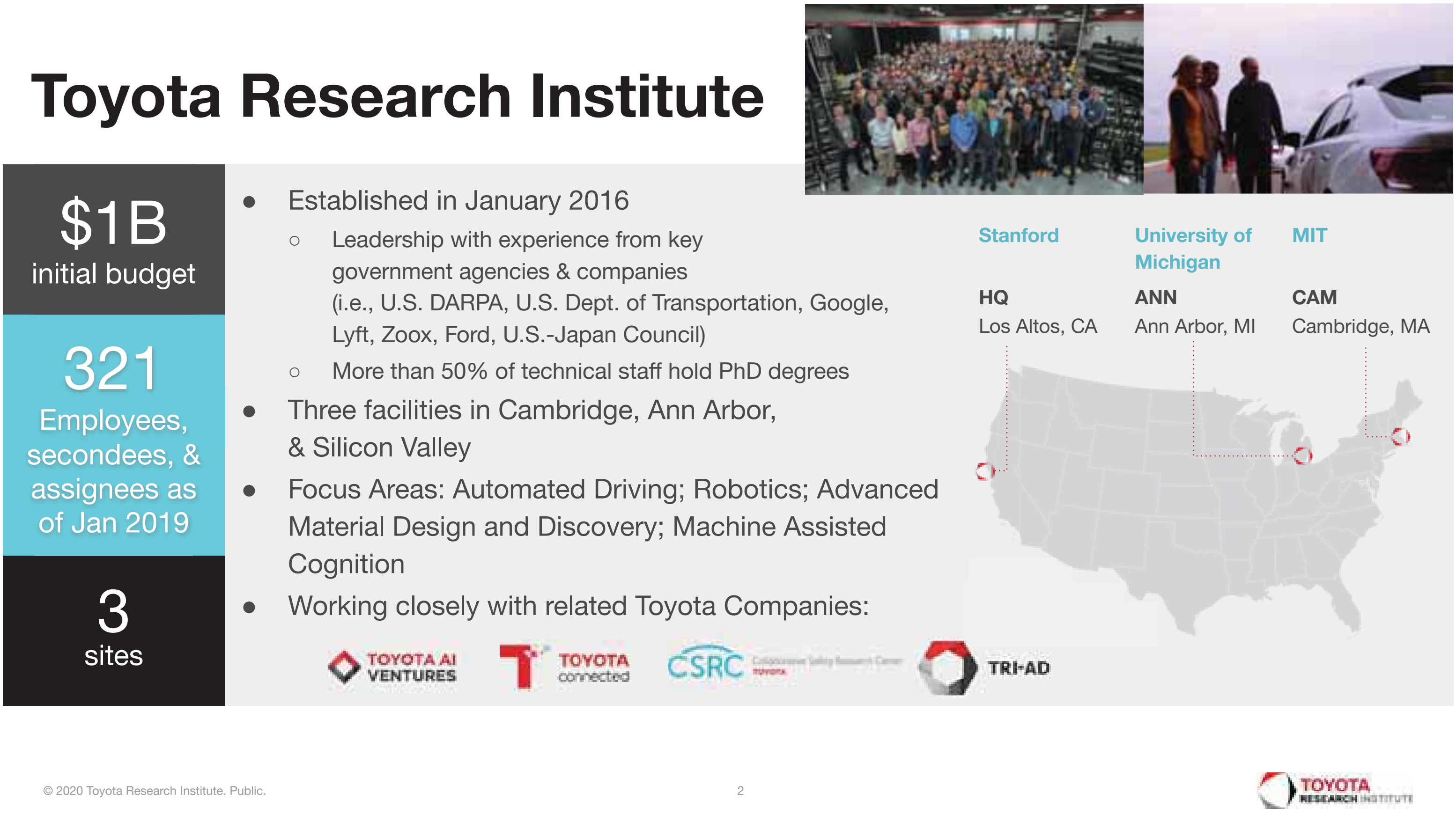
Self-Supervised Learning for Perception Tasks in Automated Driving

Dr. Wolfram Burgard

Vice President of Automated Driving Technology, Toyota Research Institute

August 2020

TOYOTA RESEARCH INSTITUTE



Toyota Research Institute

$1B
initial budget

321
Employees, secondees, & assignees as of Jan 2019

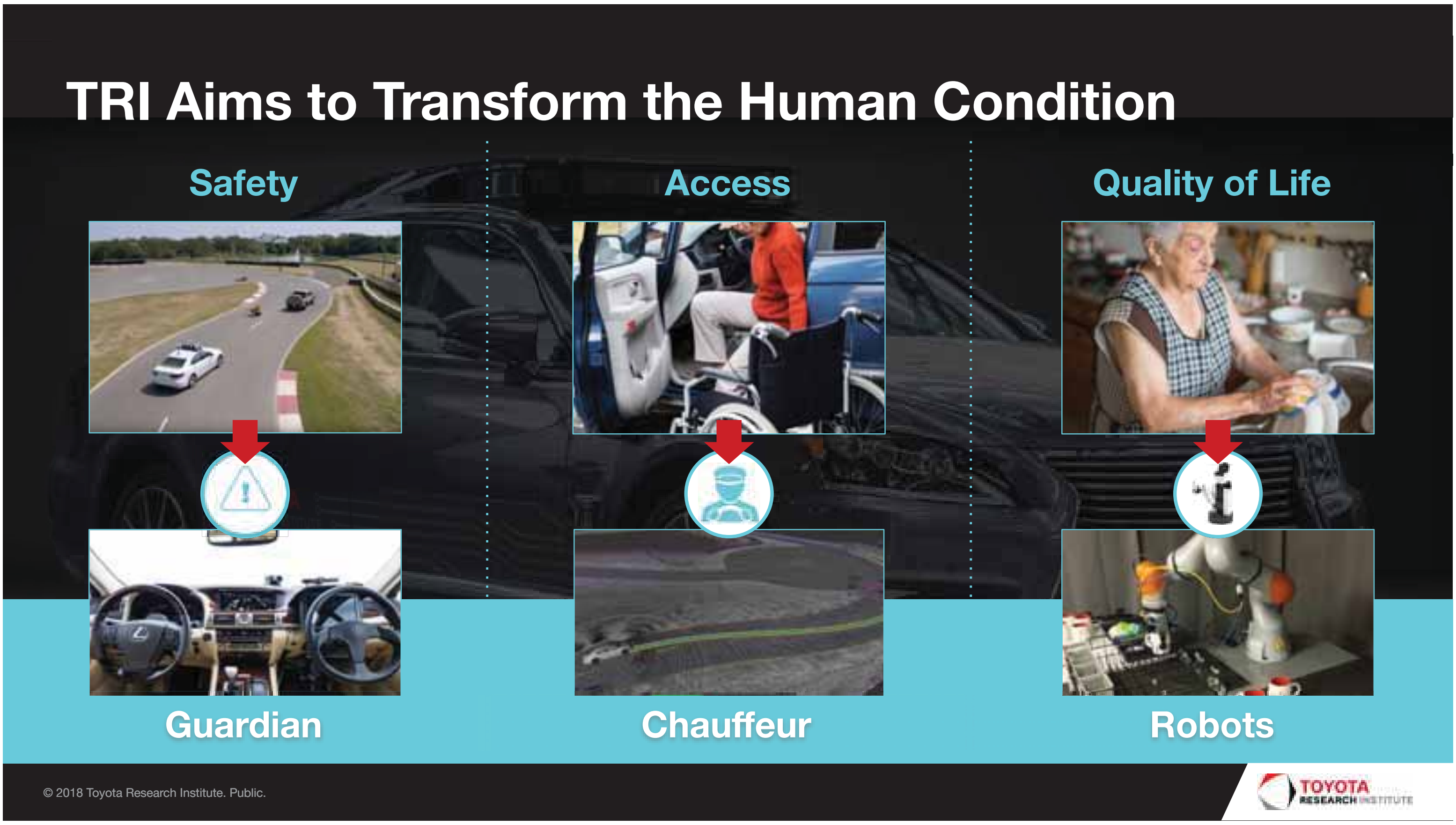
3
sites

- Established in January 2016
  - Leadership with experience from key government agencies & companies (i.e., U.S. DARPA, U.S. Dept. of Transportation, Google, Lyft, Zoox, Ford, U.S.-Japan Council)
  - More than 50% of technical staff hold PhD degrees
- Three facilities in Cambridge, Ann Arbor, & Silicon Valley
- Focus Areas: Automated Driving; Robotics; Advanced Material Design and Discovery; Machine Assisted Cognition
- Working closely with related Toyota Companies:

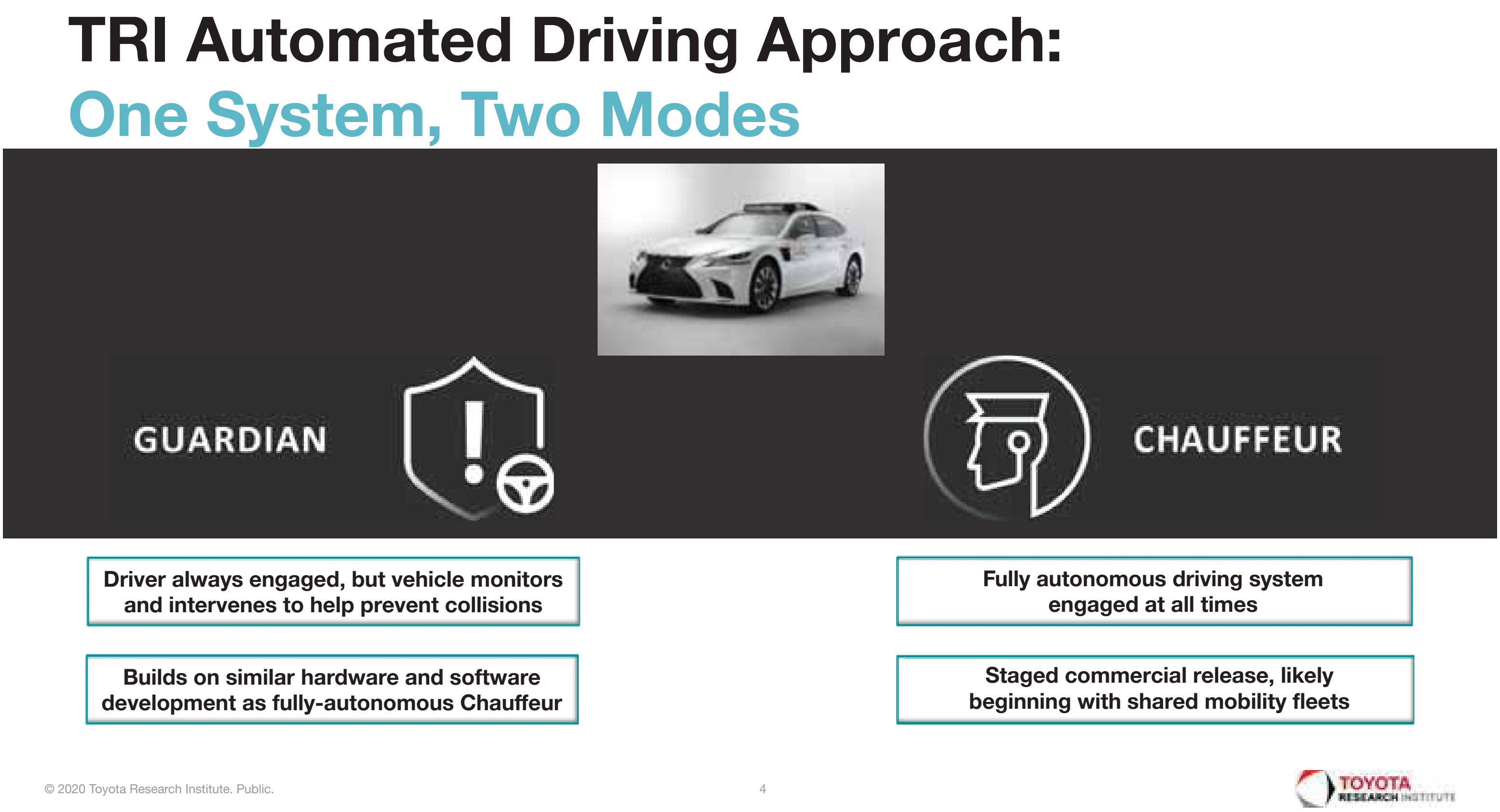
TOYOTA AI VENTURES · TOYOTA connected · CSRC · TRI-AD

Stanford	University of Michigan	MIT
HQ Los Altos, CA	ANN Ann Arbor, MI	CAM Cambridge, MA

TOYOTA RESEARCH INSTITUTE



TRI Aims to Transform the Human Condition



TRI Automated Driving Approach: One System, Two Modes

Driver always engaged, but vehicle monitors and intervenes to help prevent collisions

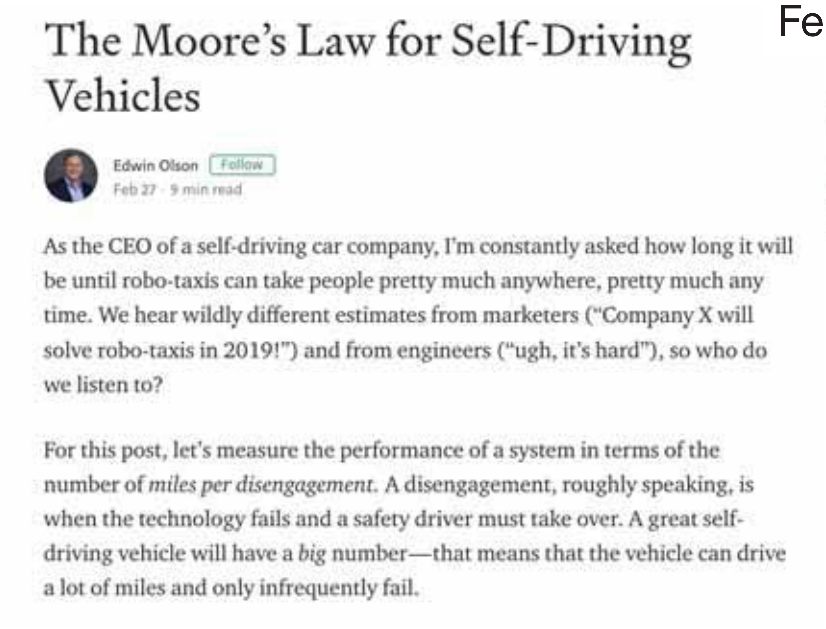
Builds on similar hardware and software development as fully-autonomous Chauffeur

Fully autonomous driving system engaged at all times

Staged commercial release, likely beginning with shared mobility fleets

Creating an Autonomous Car is Hard

Feb. 27, 2019



The Moore's Law for Self-Driving Vehicles

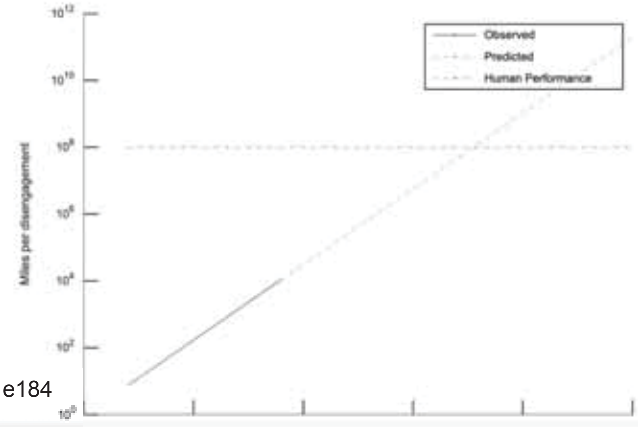
Edwin Olson Follow
Feb 27 · 9 min read

As the CEO of a self-driving car company, I'm constantly asked how long it will be until robo-taxis can take people pretty much anywhere, pretty much any time. We hear wildly different estimates from marketers ("Company X will solve robo-taxis in 2019!") and from engineers ("ugh, it's hard"), so who do we listen to?

For this post, let's measure the performance of a system in terms of the number of *miles per disengagement*. A disengagement, roughly speaking, is when the technology fails and a safety driver must take over. A great self-driving vehicle will have a *big* number—that means that the vehicle can drive a lot of miles and only infrequently fail.

... The number of miles between disengagements will double approximately every 16 months...

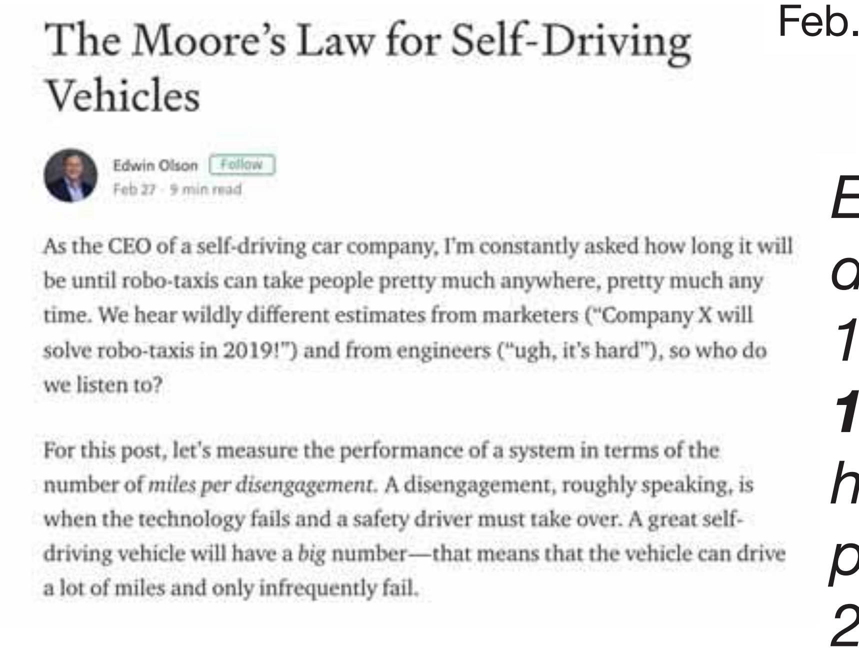
In a cosmic coincidence, the Moore's law for self-driving cars is almost the same as the Moore's law for computers—performance doubles every 16 months!



https://medium.com/may-mobility/the-moores-law-for-self-driving-vehicles-b78b8861e184

Creating an Autonomous Car is Hard

Feb. 27, 2019



The Moore's Law for Self-Driving Vehicles

Edwin Olson Follow
Feb 27 · 9 min read

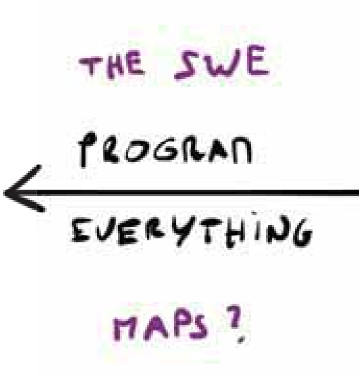
As the CEO of a self-driving car company, I'm constantly asked how long it will be until robo-taxis can take people pretty much anywhere, pretty much any time. We hear wildly different estimates from marketers ("Company X will solve robo-taxis in 2019!") and from engineers ("ugh, it's hard"), so who do we listen to?

For this post, let's measure the performance of a system in terms of the number of *miles per disengagement*. A disengagement, roughly speaking, is when the technology fails and a safety driver must take over. A great self-driving vehicle will have a *big* number—that means that the vehicle can drive a lot of miles and only infrequently fail.

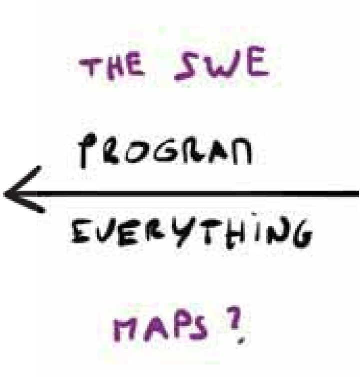
*Even with performance doubling every 16 months, it will take **16 years** to reach human levels of performance — that's 2035.*

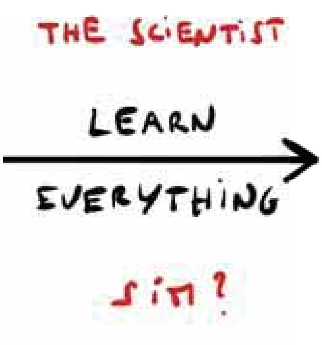
https://medium.com/may-mobility/the-moores-law-for-self-driving-vehicles-b78b8861e184

World-scale Autonomy?

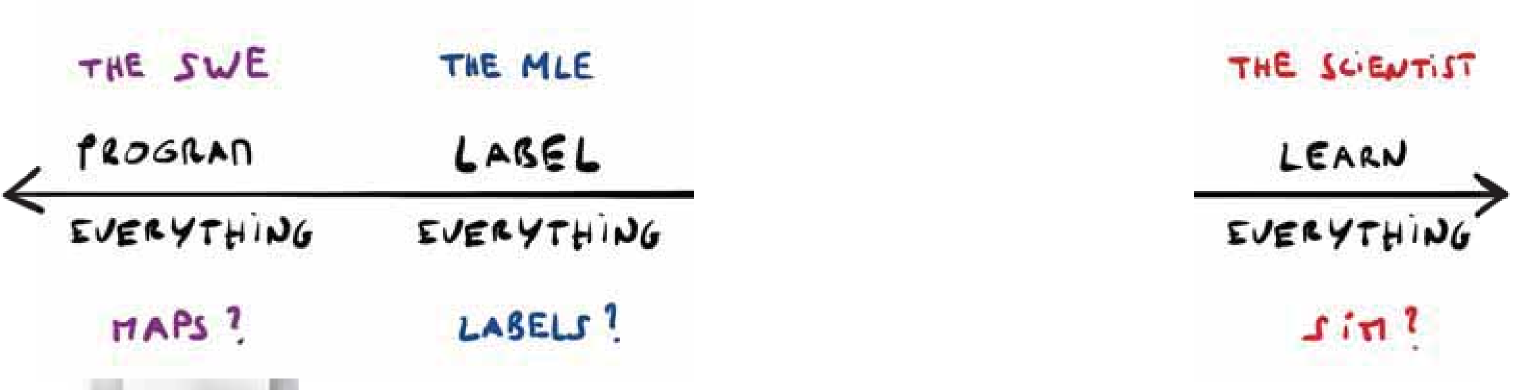


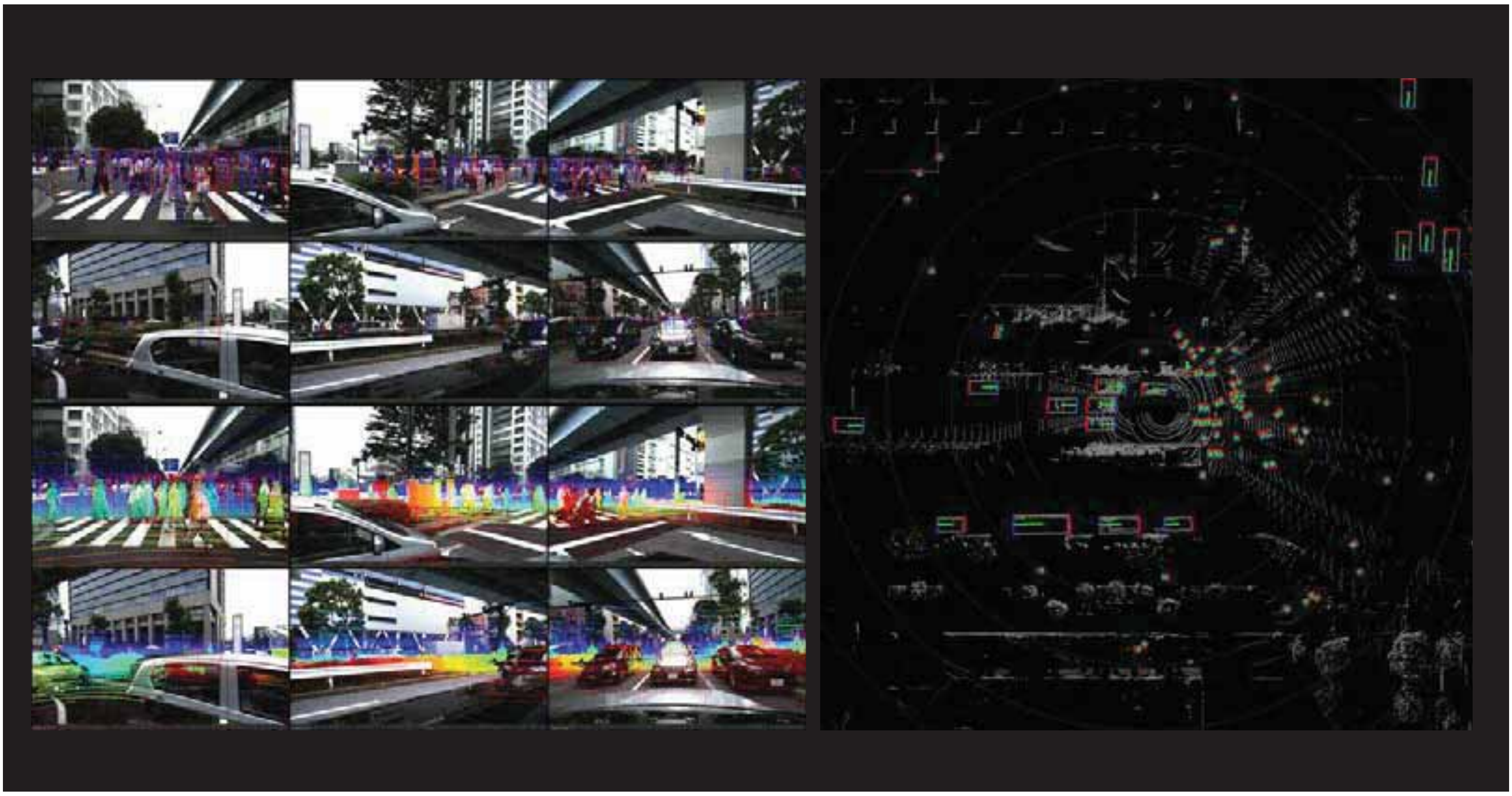
World-scale Autonomy?



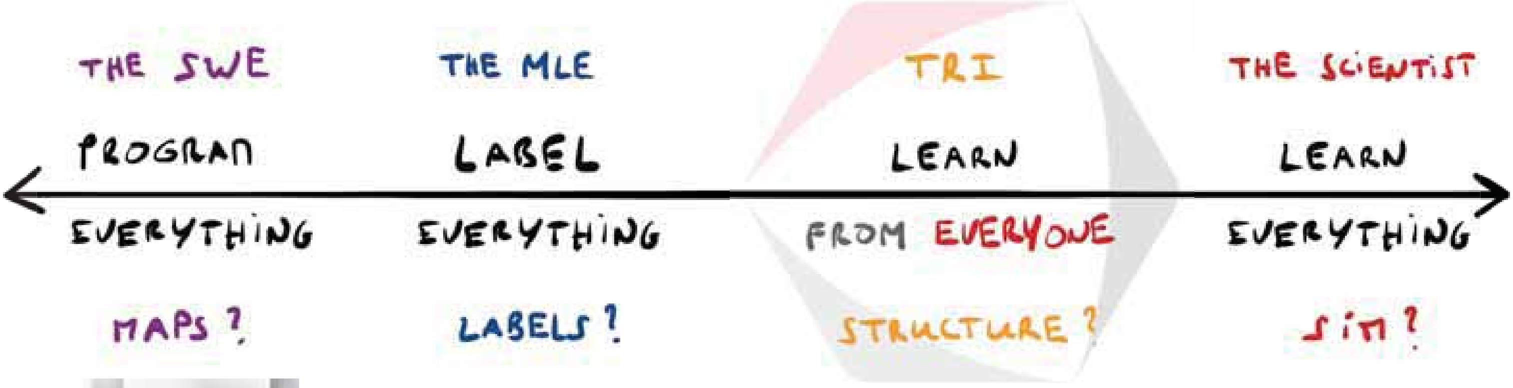


World-scale Autonomy?





World-scale Autonomy?



Toyota's Strategic Data Advantage

- Unprecedented scale of data
  - Largest sensor fleet
  - Cover all US roads in under a day: Multiple times!
- Effective use of data
  - Learning from everything is infeasible!
  - Learn from unbiased, diverse and representative data
  - Leverage large volumes of unlabeled, structured data
  - **Data curation, querying, and synthesis**
- Our strategic focus
  - Supervised learning + **Self-supervised learning** from large volumes of structured and unlabeled data

We need to be smart about drinking from the data firehose

Self-Supervised Learning, Learning with Maps,
Transfer Learning, Representation Learning

Agenda

- **Self-Supervised Learning: SuperDepth**
- **Self-Supervised Pseudo-Lidar Networks**
- **Real-time Panoptic Segmentation**

Publication

SuperDepth: Self-Supervised, Super-Resolved Monocular Depth Estimation,

S. Pillai, R. Ambrus, A. Gaidon,

ICRA 2019

SuperDepth: Self-Supervised, Super-Resolved Monocular Depth Estimation

Sudeep Pillai, Rares Ambrus, Adrien Gaidon
Toyota Research Institute (TRI)

arXiv:1810.01849v1 [cs.CV] 3 Oct 2018

[illegible]

I. Introduction

[illegible]

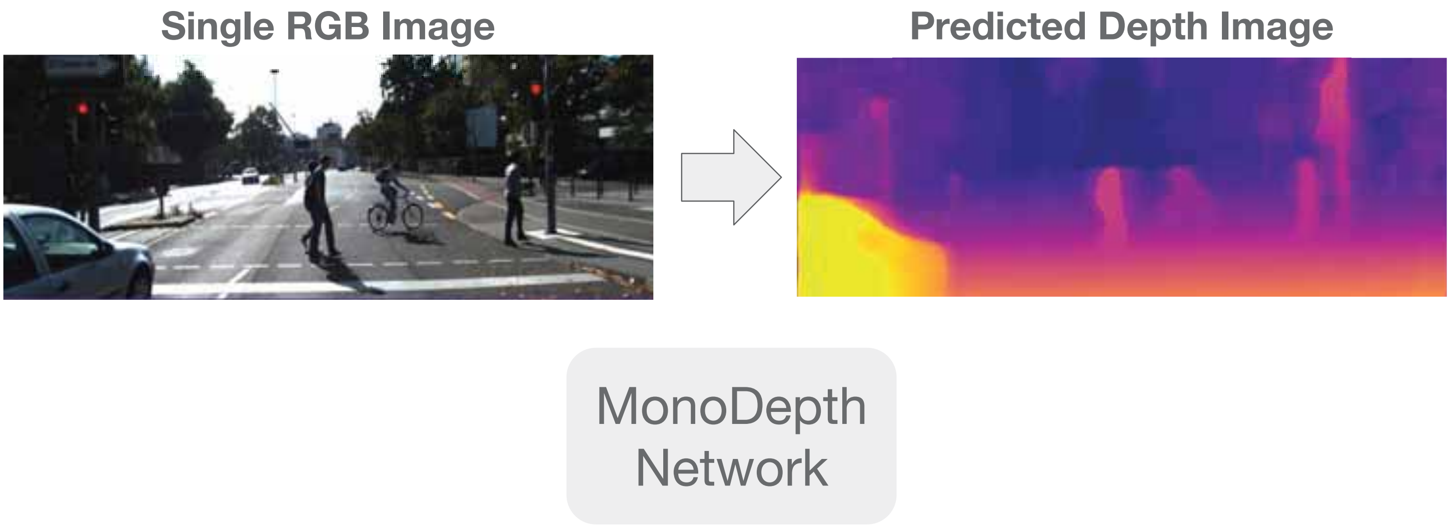
Self-Supervised Learning at Toyota-scale

- **SuperDepth: Self-Supervised Monocular Depth**
  - Exploit **large volumes** of **unlabeled**, **structured** camera data
  - Training **only** requires **unlabeled driving video data!**
- Why MonoDepth?
  - LiDARs are expensive and bulky
  - Cameras
    - Rich semantic and geometric sensing
    - Ubiquitous (2019 Toyota models)

Toyota Safety Sense 2.0
Camera

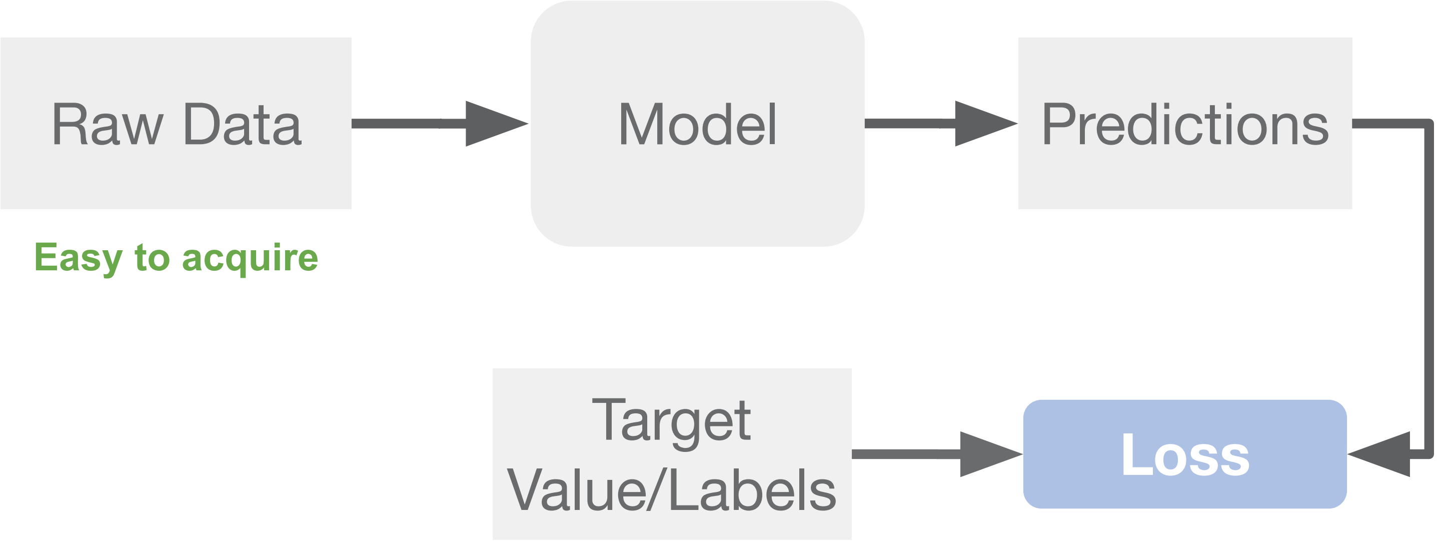
© 2018 Toyota Research Institute.
Confidential. Do not distribute.

Monocular Depth Estimation



© 2018 Toyota Research Institute. Public.

Supervised Learning



Raw Data

Model

Predictions

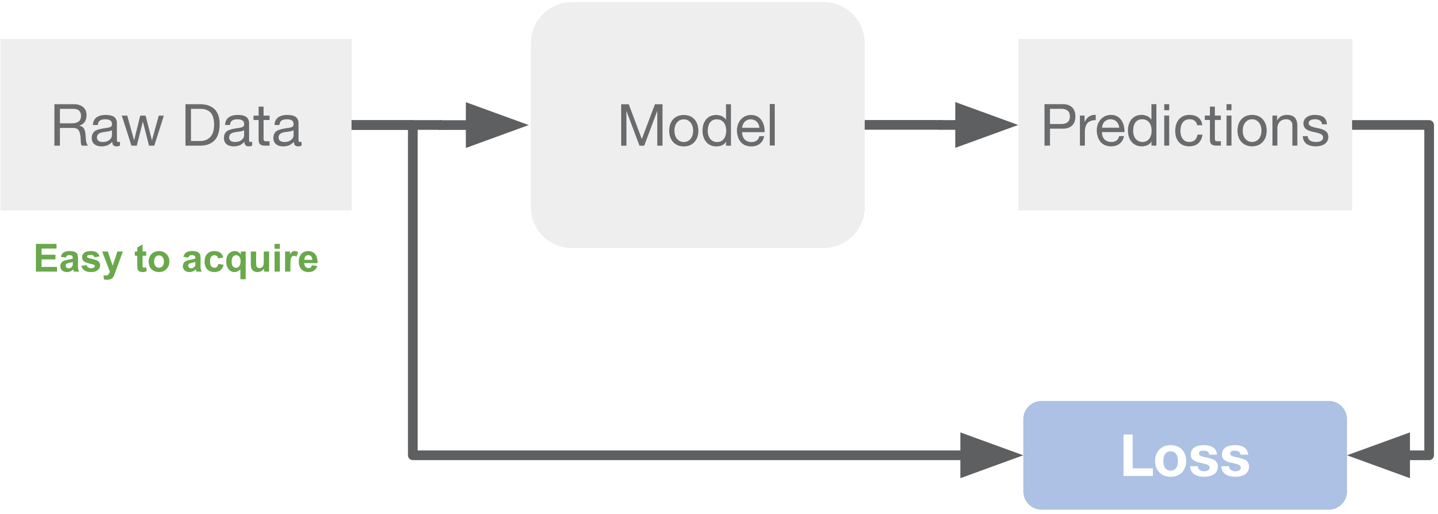
Easy to acquire

Target Value/Labels

Loss

Expensive / Difficult to acquire

Self-Supervised Learning



Raw Data

Model

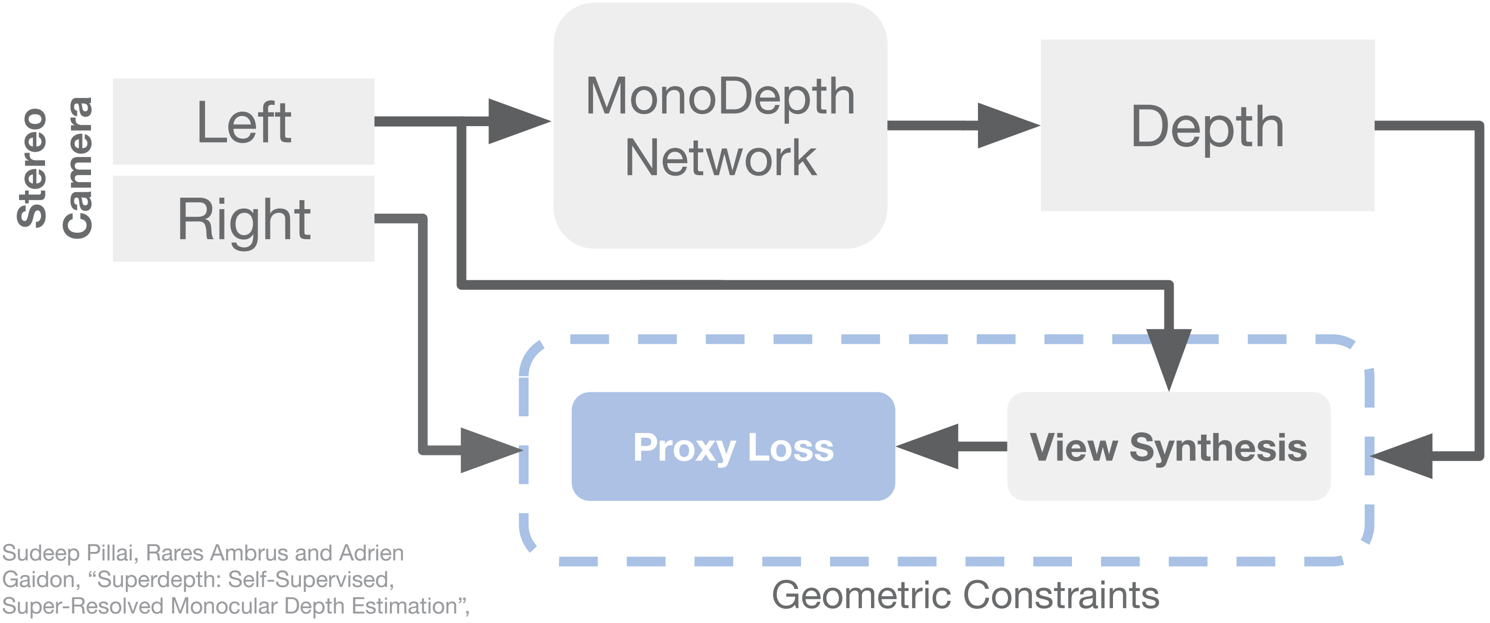
Predictions

Easy to acquire

Loss

Prior Knowledge

Self-Supervised Monocular Depth



Sudeep Pillai, Rares Ambrus and Adrien Gaidon, “Superdepth: Self-Supervised, Super-Resolved Monocular Depth Estimation”, ICRA 2019

Self-Supervised Depth Learning Objective

$$\hat{\theta_D} = \arg\min_{\theta_D} \sum_{s\in S} \mathcal{L}_D(I_t, \hat{I}_t; \theta_D)$$

Depth Model Parameters

$$\mathcal{L}_D(I_t, \hat{I}_t) = \mathcal{L}_p(I_t, \hat{I}_t) + \lambda_1 \; \mathcal{L}_s(I_t) + \lambda_2 \; \mathcal{L}_o(I_t)$$

Photometric loss via view-synthesis

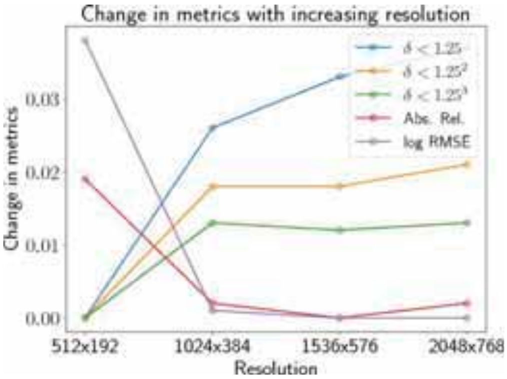
Depth Regularization (edge-aware depth smoothing)

Occlusion Regularization

Photometric Loss ++

- Multi-scale photometric loss is **limited** by resolution
- Super-resolve disparities → **synthesize at high resolutions**

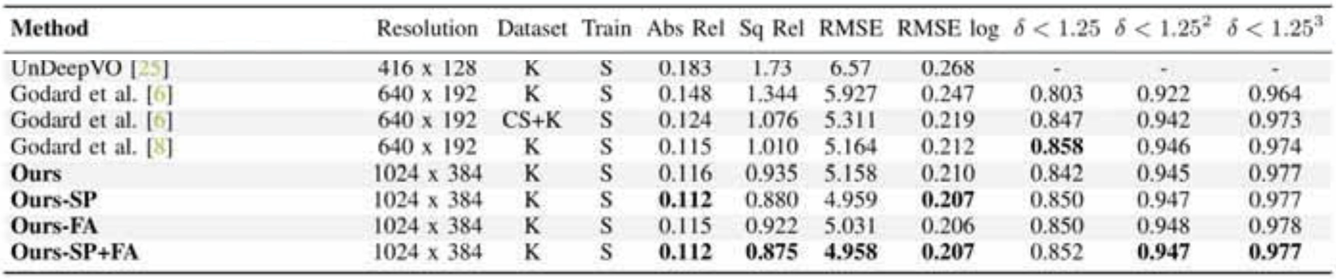
Resolution Matters for View Synthesis!



Depth estimation accuracy **increases** with increasing high-resolution

Abs. Rel, and log RMSE (lower is better)

Disparity Estimation Performance

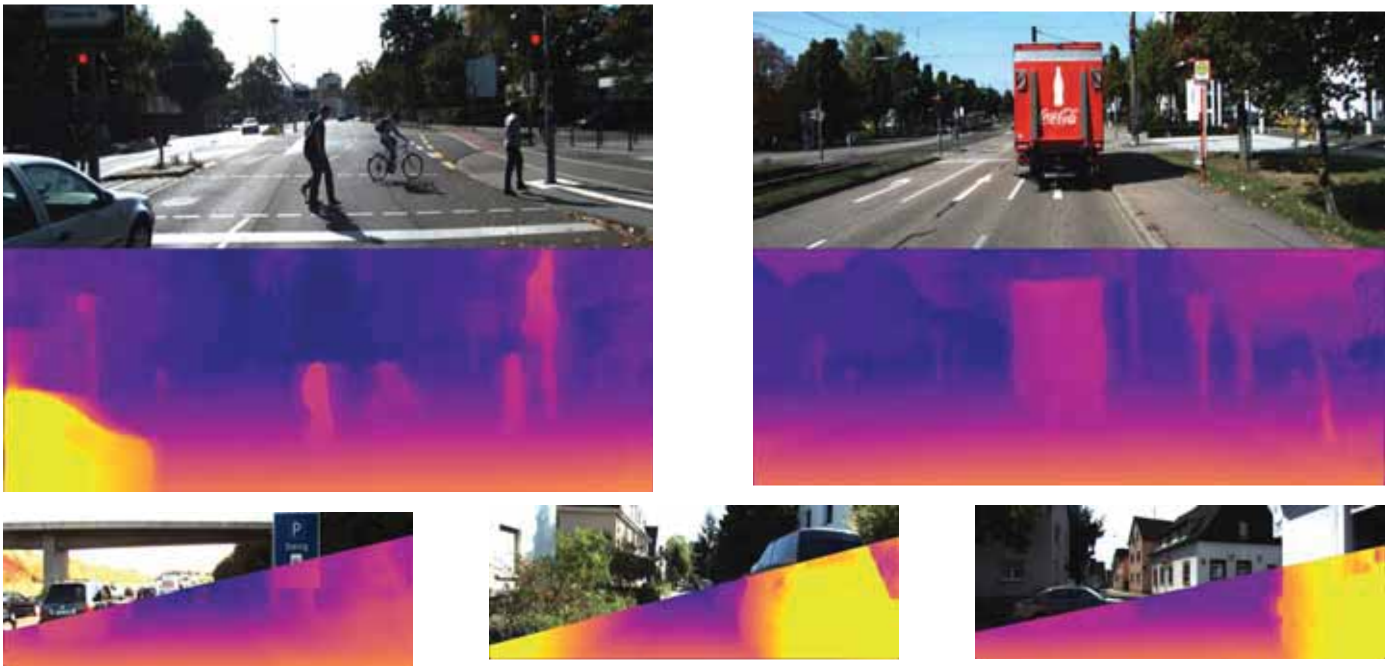


Method	Resolution	Dataset	Train	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
UnDeepVO [25]	416 x 128	K	S	0.183	1.73	6.57	0.268	-	-	-
Godard et al. [6]	640 x 192	K	S	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Godard et al. [6]	640 x 192	CS+K	S	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Godard et al. [8]	640 x 192	K	S	0.115	1.010	5.164	0.212	**0.858**	0.946	0.974
Ours	1024 x 384	K	S	0.116	0.935	5.158	0.210	0.842	0.945	0.977
Ours-SP	1024 x 384	K	S	**0.112**	0.880	4.959	**0.207**	0.850	0.947	0.977
Ours-FA	1024 x 384	K	S	0.115	0.922	5.031	0.206	0.850	0.948	0.978
Ours-SP+FA	1024 x 384	K	S	**0.112**	**0.875**	**4.958**	**0.207**	0.852	**0.947**	**0.977**

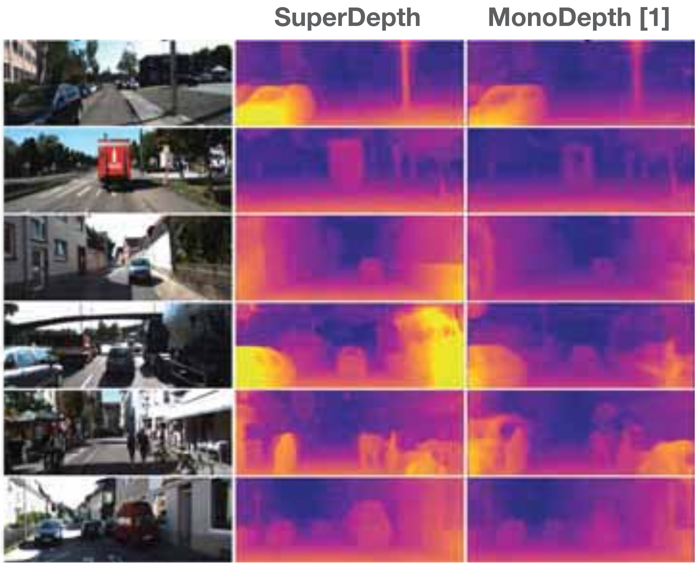
Depth Estimation Results on the KITTI 2015 Benchmark

Sub-pixel convolutions (**SP**), Differentiable Flip Augmentation (**FA**)

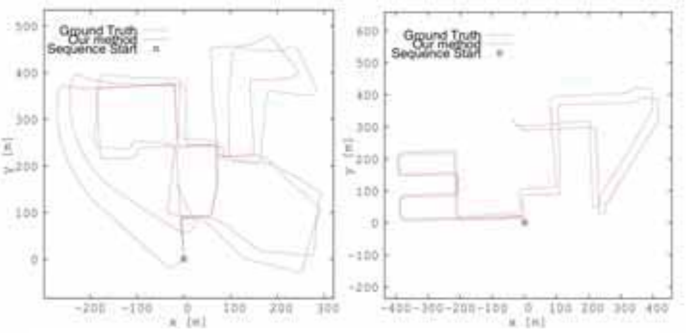
Qualitative MonoDepth Performance



Qualitative Comparison to State-of-the-Art



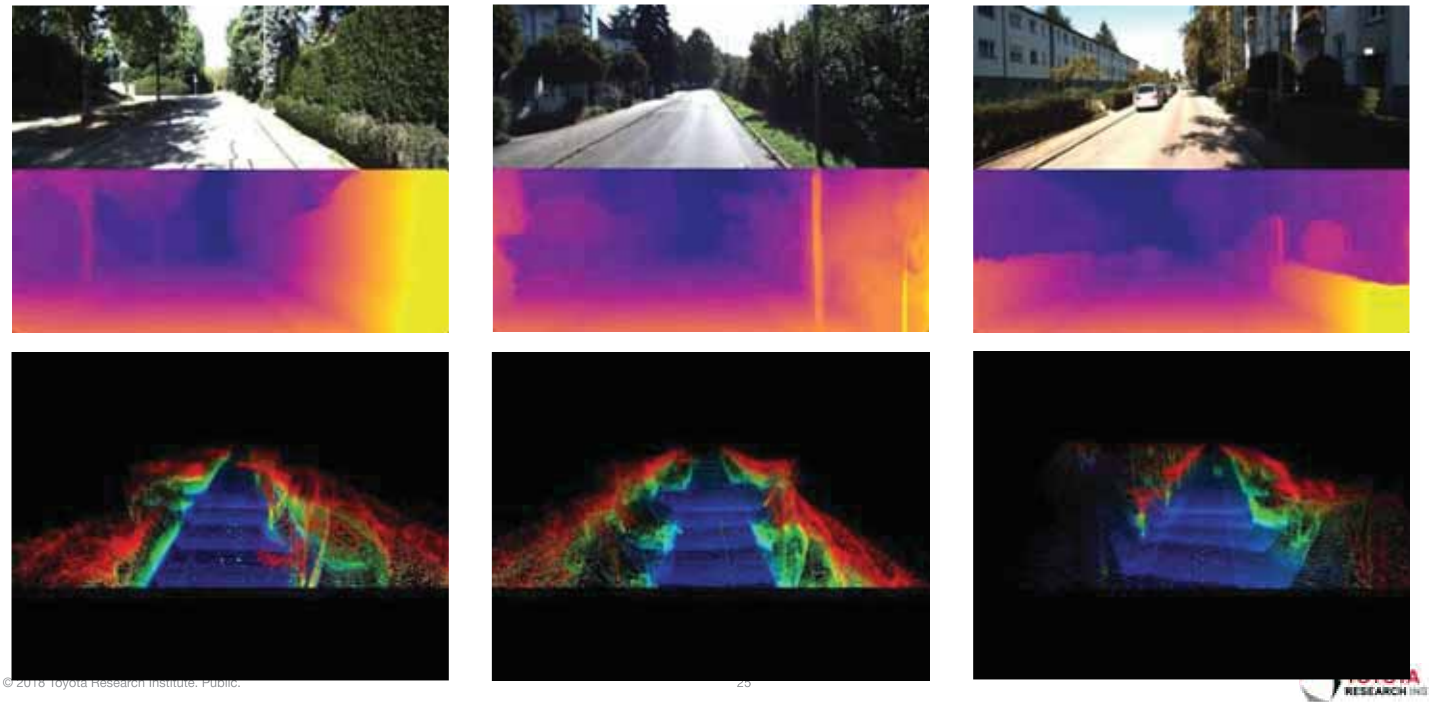
SuperDepth reconstruction is able to capture **fine details**, and **boundaries**



Bonus: We can also recover long-term, **scale-aware camera ego-motion from a single camera!**

[1] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” CVPR, 2017

Dense Monocular 3D Reconstruction



Agenda

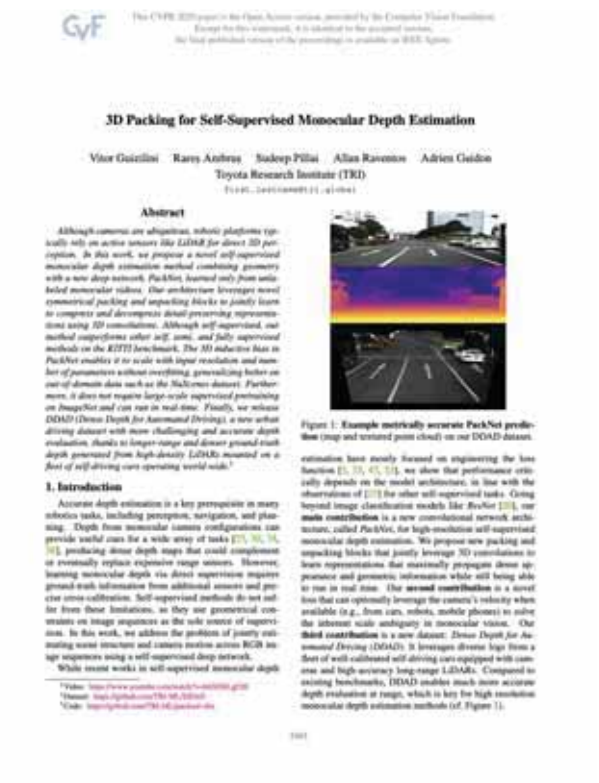
- Self-Supervised Learning: SuperDepth
- Self-Supervised Pseudo-Lidar Networks
- Real-time Panoptic Segmentation

Publication

3D Packing for
Self-Supervised Monocular
Depth Estimation,

V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, A. Gaidon,

CVPR 2020, oral presentation



3D Packing for Self-Supervised Monocular Depth Estimation

Vitor Guizilini Rareș Ambruș Sudeep Pillai Allan Raventos Adrien Gaidon
Toyota Research Institute (TRI)
first.lastname@tri.global

Abstract

Although cameras are ubiquitous, robotic platforms typically rely on active sensors like LiDAR for direct 3D perception. In this work, we propose a novel self-supervised monocular depth estimation method combining geometry with a new deep network, PackNet, learned only from unlabeled monocular videos. Our architecture leverages novel symmetrical packing and unpacking blocks to jointly learn to compress and decompress detail-preserving representations using 3D convolutions. Although self-supervised, our method outperforms other self, semi, and fully supervised methods on the KITTI benchmark. The 3D inductive bias in PackNet enables it to scale with input resolution and number of parameters without overfitting, generalizing better on out-of-domain data such as the NuScenes dataset. Furthermore, it does not require large-scale supervised pretraining on ImageNet and can run in real-time. Finally, we release DDAD (Dense Depth for Automated Driving), a new urban driving dataset with more challenging and accurate depth evaluation, thanks to longer-range and denser ground-truth depth generated from high-density LiDARs mounted on a fleet of self-driving cars operating world-wide.[1]

1. Introduction

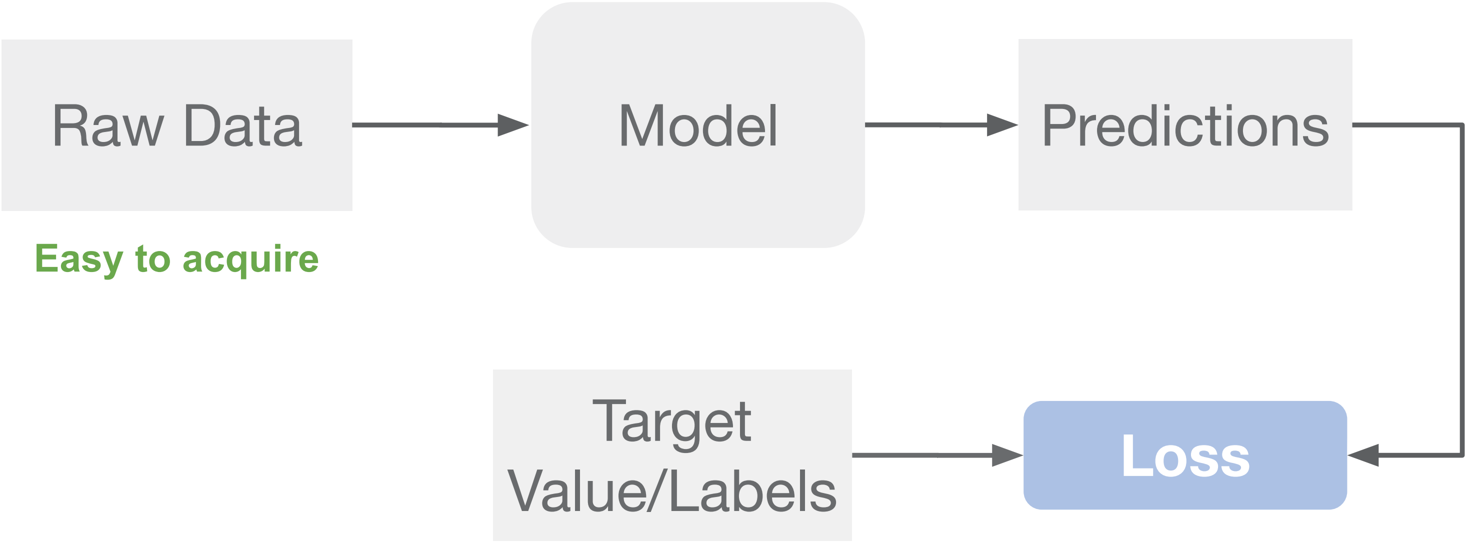
Accurate depth estimation is a key prerequisite in many robotics tasks, including perception, navigation, and planning. Depth from monocular camera configurations can provide useful cues for a wide array of tasks [illegible], producing dense depth maps that could complement or eventually replace expensive range sensors. However, learning monocular depth via direct supervision requires ground-truth information from additional sensors and precise cross-calibration. Self-supervised methods do not suffer from these limitations, as they use geometrical constraints on image sequences as the sole source of supervision. In this work, we address the problem of jointly estimating scene structure and camera motion across RGB image sequences using a self-supervised deep network.

While recent works in self-supervised monocular depth estimation have mostly focused on engineering the loss function [illegible], we show that performance critically depends on the model architecture, in line with the observations of [illegible] for other self-supervised tasks. Going beyond image classification models like *ResNet* [illegible], our **main contribution** is a new convolutional network architecture, called *PackNet*, for high-resolution self-supervised monocular depth estimation. We propose new packing and unpacking blocks that jointly leverage 3D convolutions to learn representations that maximally propagate dense appearance and geometric information while still being able to run in real time. Our **second contribution** is a novel loss that can optionally leverage the camera's velocity when available (e.g., from cars, robots, mobile phones) to solve the inherent scale ambiguity in monocular vision. Our **third contribution** is a new dataset: *Dense Depth for Automated Driving* (DDAD). It leverages diverse logs from a fleet of well-calibrated self-driving cars equipped with cameras and high-accuracy long-range LiDARs. Compared to existing benchmarks, DDAD enables much more accurate depth evaluation at range, which is key for high resolution monocular depth estimation methods (cf. Figure 1).

Figure 1: **Example metrically accurate PackNet prediction** (map and textured point cloud) on our DDAD dataset.

[1] Video: [illegible]
Dataset: https://github.com/TRI-ML/DDAD
Code: https://github.com/TRI-ML/packnet-sfm

Supervised Learning



Raw Data

Model

Predictions

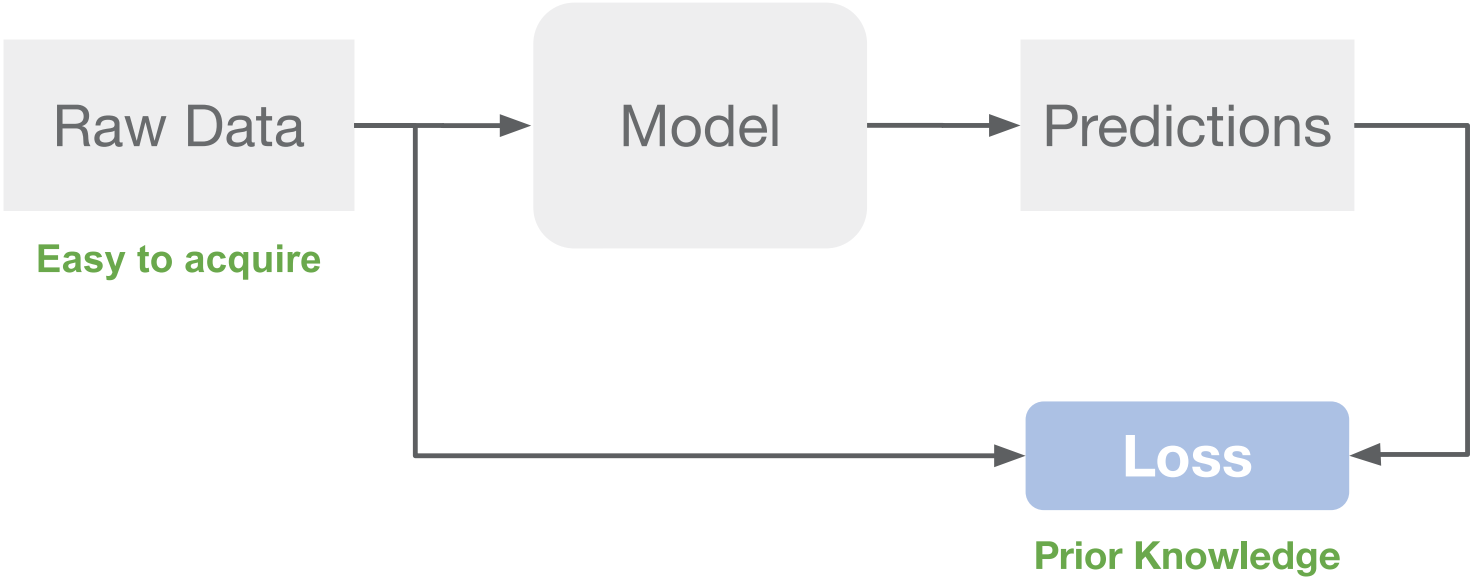
Target Value/Labels

Loss

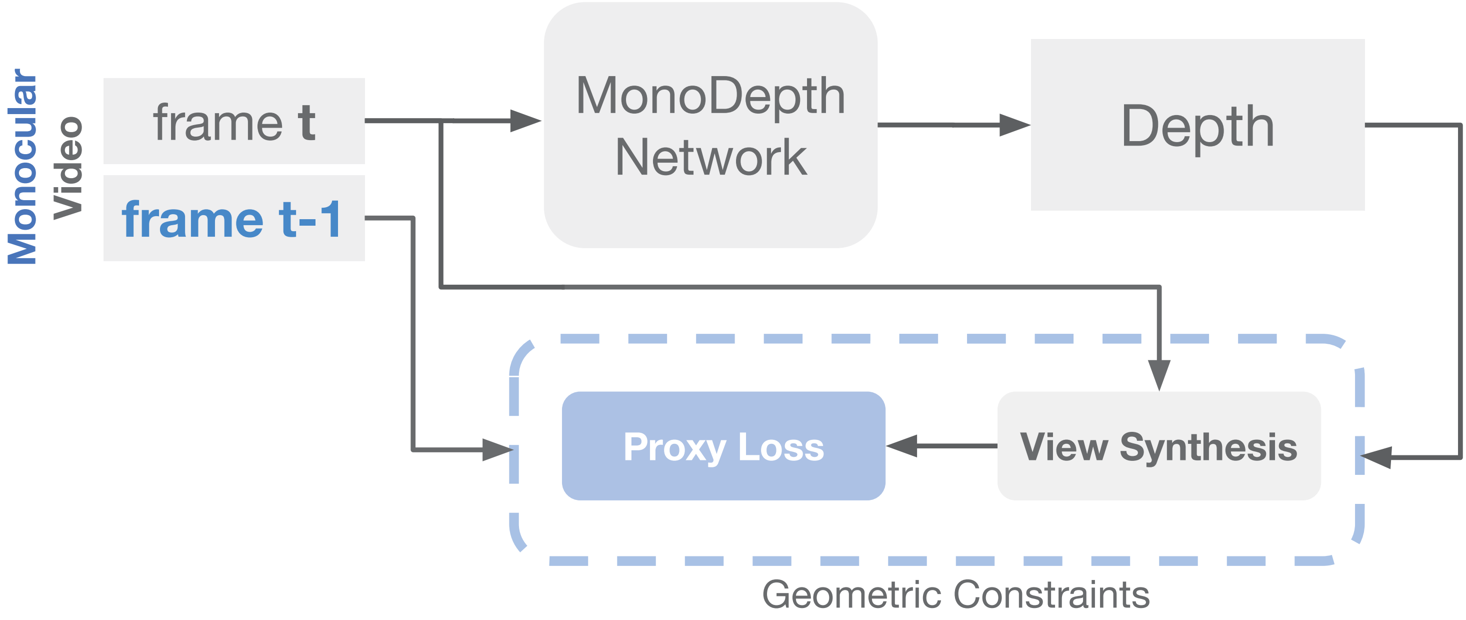
Easy to acquire

Expensive / Difficult to acquire

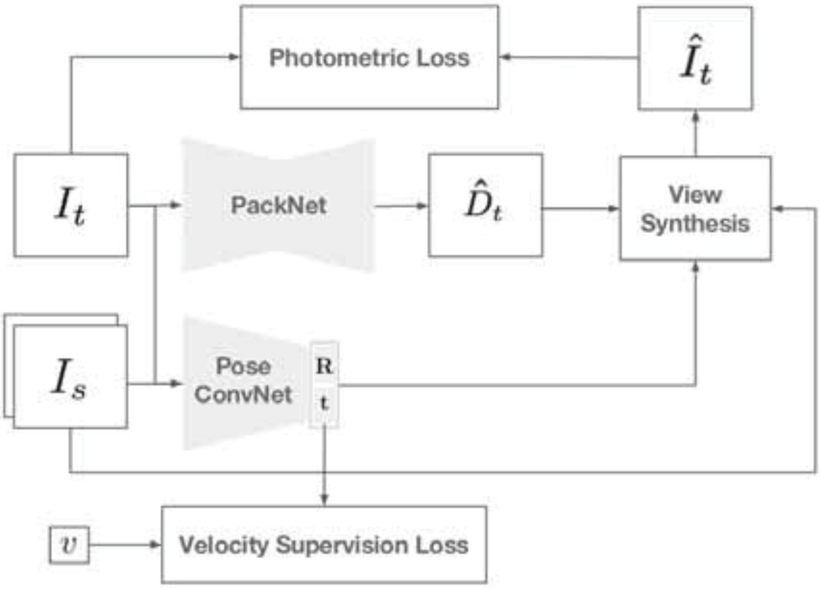
Self-Supervised Learning



Self-Supervised Structure-from-Motion (SfM)



Self-Supervised Structure-from-Motion (SfM)



Photometric Loss

$\hat{I}_t$

I_t

PackNet

$\hat{D}_t$

View Synthesis

I_s

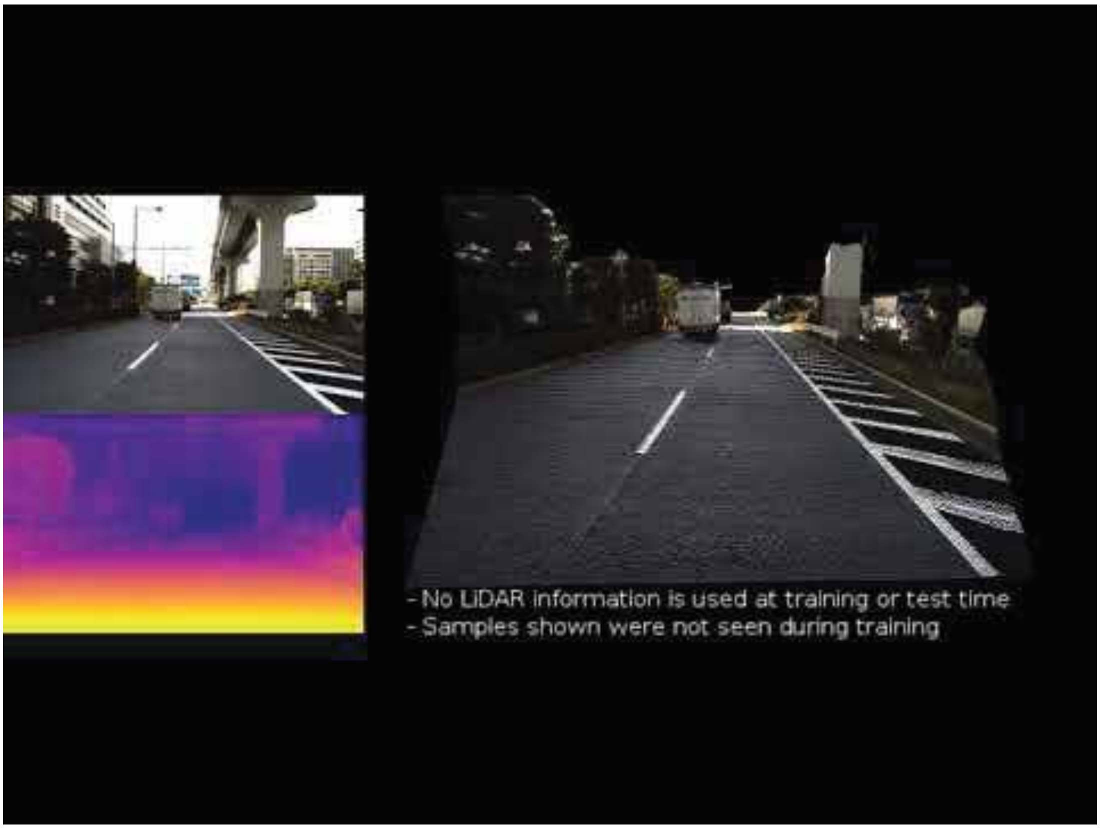
Pose ConvNet

R

t

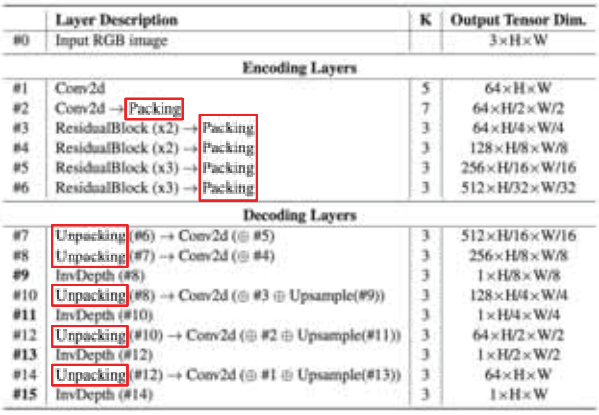
v

Velocity Supervision Loss

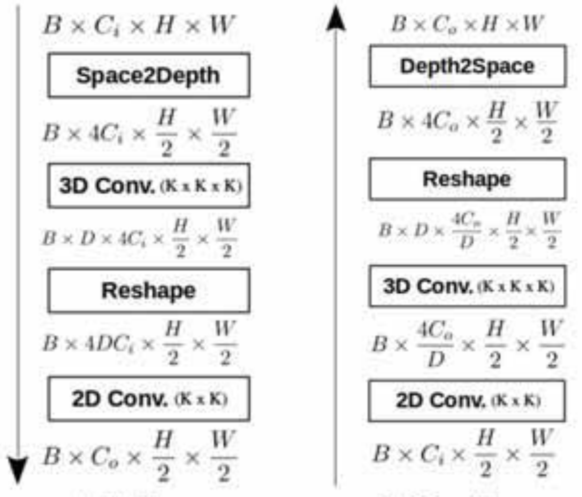


- No LiDAR information is used at training or test time
- Samples shown were not seen during training

PackNet: Pack it, don't pool it



	Layer Description	K	Output Tensor Dim.
#0	Input RGB image		3×H×W
	Encoding Layers		
#1	Conv2d	5	64×H×W
#2	Conv2d → Packing	7	64×H/2×W/2
#3	ResidualBlock (x2) → Packing	3	64×H/4×W/4
#4	ResidualBlock (x2) → Packing	3	128×H/8×W/8
#5	ResidualBlock (x3) → Packing	3	256×H/16×W/16
#6	ResidualBlock (x3) → Packing	3	512×H/32×W/32
	Decoding Layers		
#7	Unpacking (#6) → Conv2d (⊕ #5)	3	512×H/16×W/16
#8	Unpacking (#7) → Conv2d (⊕ #4)	3	256×H/8×W/8
#9	InvDepth (#8)	3	1×H/8×W/8
#10	Unpacking (#8) → Conv2d (⊕ #3 ⊕ Upsample(#9))	3	128×H/4×W/4
#11	InvDepth (#10)	3	1×H/4×W/4
#12	Unpacking (#10) → Conv2d (⊕ #2 ⊕ Upsample(#11))	3	64×H/2×W/2
#13	InvDepth (#12)	3	1×H/2×W/2
#14	Unpacking (#12) → Conv2d (⊕ #1 ⊕ Upsample(#13))	3	64×H×W
#15	InvDepth (#14)	3	1×H×W


$B \times C_i \times H \times W$
Space2Depth
$B \times 4C_i \times \frac{H}{2} \times \frac{W}{2}$
3D Conv. (K x K x K)
$B \times D \times 4C_i \times \frac{H}{2} \times \frac{W}{2}$
Reshape
$B \times 4DC_i \times \frac{H}{2} \times \frac{W}{2}$
2D Conv. (K x K)
$B \times C_o \times \frac{H}{2} \times \frac{W}{2}$
$B \times C_o \times H \times W$
Depth2Space
$B \times 4C_o \times \frac{H}{2} \times \frac{W}{2}$
Reshape
$B \times D \times \frac{4C_o}{D} \times \frac{H}{2} \times \frac{W}{2}$
3D Conv. (K x K x K)
$B \times \frac{4C_o}{D} \times \frac{H}{2} \times \frac{W}{2}$
2D Conv. (K x K)
$B \times C_i \times \frac{H}{2} \times \frac{W}{2}$

(a) Packing

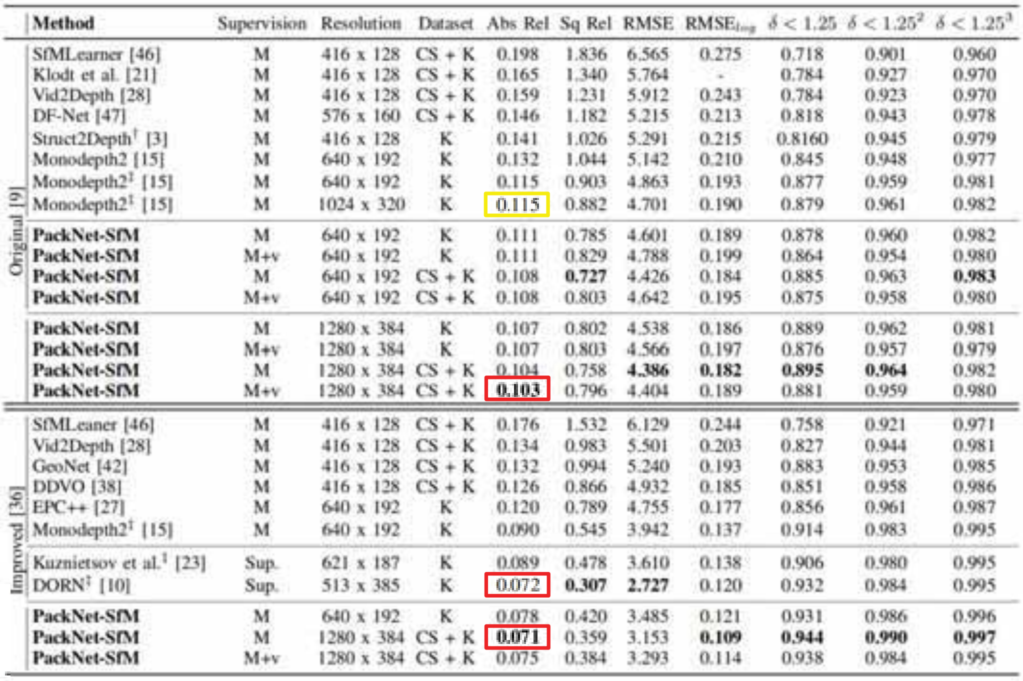
(b) Unpacking

(a) Input Image

(b) Max Pooling + Bilinear Upsample

(c) Pack + Unpack

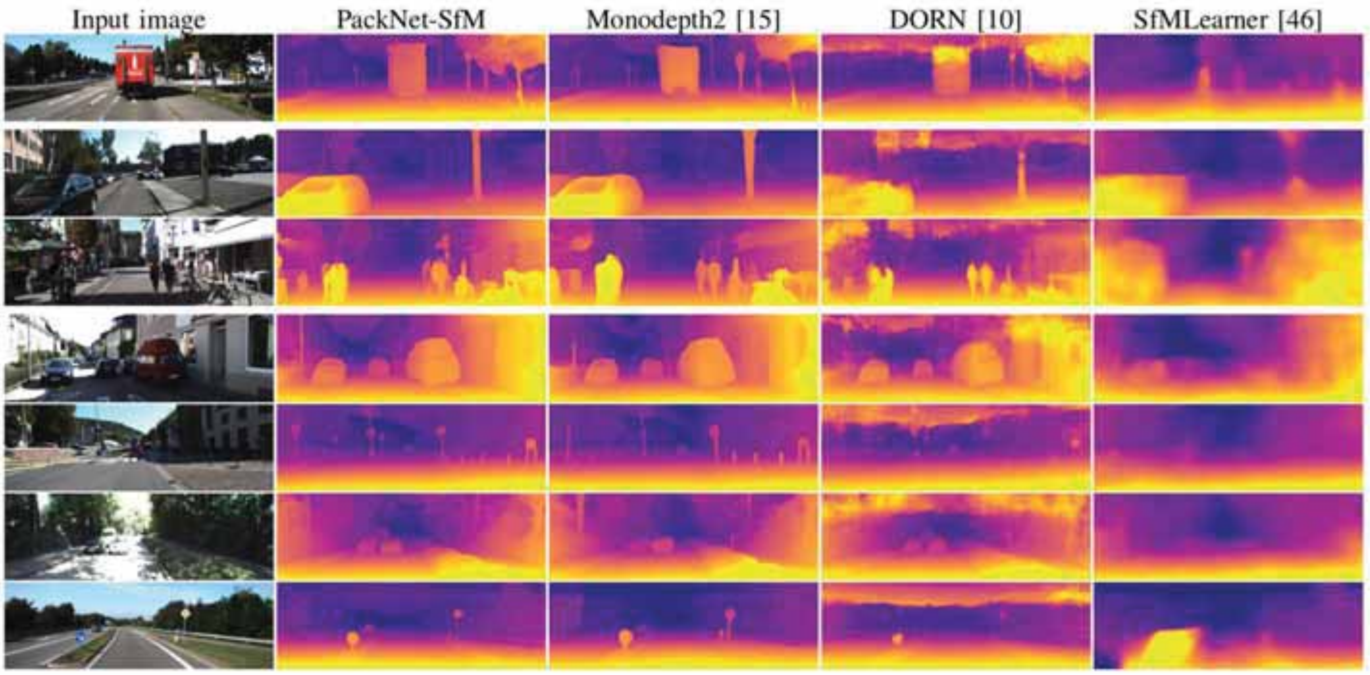
Experimental Results (KITTI)



	Method	Supervision	Resolution	Dataset	Abs Rel	Sq Rel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Original [9]	SfMLearner [46]	M	416 x 128	CS + K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
	Klodt et al. [21]	M	416 x 128	CS + K	0.165	1.340	5.764	-	0.784	0.927	0.970
	Vid2Depth [28]	M	416 x 128	CS + K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
	DF-Net [47]	M	576 x 160	CS + K	0.146	1.182	5.215	0.213	0.818	0.943	0.978
	Struct2Depth† [3]	M	416 x 128	K	0.141	1.026	5.291	0.215	0.8160	0.945	0.979
	Monodepth2 [15]	M	640 x 192	K	0.132	1.044	5.142	0.210	0.845	0.948	0.977
	Monodepth2‡ [15]	M	640 x 192	K	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	Monodepth2‡ [15]	M	1024 x 320	K	0.115	0.882	4.701	0.190	0.879	0.961	0.982
	PackNet-SfM	M	640 x 192	K	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	PackNet-SfM	M+v	640 x 192	K	0.111	0.829	4.788	0.199	0.864	0.954	0.980
	PackNet-SfM	M	640 x 192	CS + K	0.108	**0.727**	4.426	0.184	0.885	0.963	**0.983**
	PackNet-SfM	M+v	640 x 192	CS + K	0.108	0.803	4.642	0.195	0.875	0.958	0.980
	PackNet-SfM	M	1280 x 384	K	0.107	0.802	4.538	0.186	0.889	0.962	0.981
	PackNet-SfM	M+v	1280 x 384	K	0.107	0.803	4.566	0.197	0.876	0.957	0.979
	PackNet-SfM	M	1280 x 384	CS + K	0.104	0.758	**4.386**	**0.182**	**0.895**	**0.964**	0.982
	PackNet-SfM	M+v	1280 x 384	CS + K	**0.103**	0.796	4.404	0.189	0.881	0.959	0.980
Improved [36]	SfMLeaner [46]	M	416 x 128	CS + K	0.176	1.532	6.129	0.244	0.758	0.921	0.971
	Vid2Depth [28]	M	416 x 128	CS + K	0.134	0.983	5.501	0.203	0.827	0.944	0.981
	GeoNet [42]	M	416 x 128	CS + K	0.132	0.994	5.240	0.193	0.883	0.953	0.985
	DDVO [38]	M	416 x 128	CS + K	0.126	0.866	4.932	0.185	0.851	0.958	0.986
	EPC++ [27]	M	640 x 192	K	0.120	0.789	4.755	0.177	0.856	0.961	0.987
	Monodepth2‡ [15]	M	640 x 192	K	0.090	0.545	3.942	0.137	0.914	0.983	0.995
	Kuznietsov et al.‡ [23]	Sup.	621 x 187	K	0.089	0.478	3.610	0.138	0.906	0.980	0.995
	DORN‡ [10]	Sup.	513 x 385	K	0.072	**0.307**	**2.727**	0.120	0.932	0.984	0.995
	PackNet-SfM	M	640 x 192	K	0.078	0.420	3.485	0.121	0.931	0.986	0.996
	PackNet-SfM	M	1280 x 384	CS + K	**0.071**	0.359	3.153	**0.109**	**0.944**	**0.990**	**0.997**
	PackNet-SfM	M+v	1280 x 384	CS + K	0.075	0.384	3.293	0.114	0.938	0.984	0.995

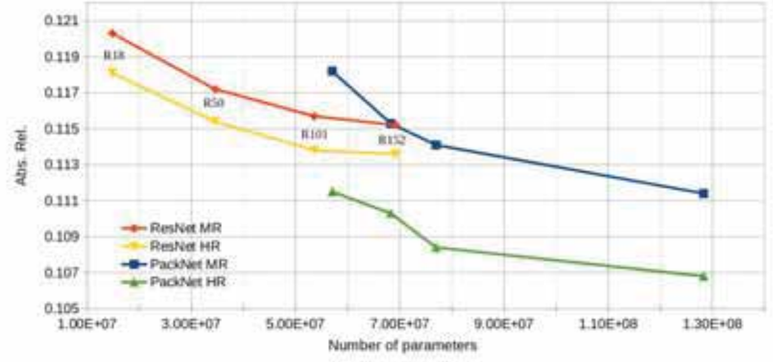
Self-sup. better than sup!

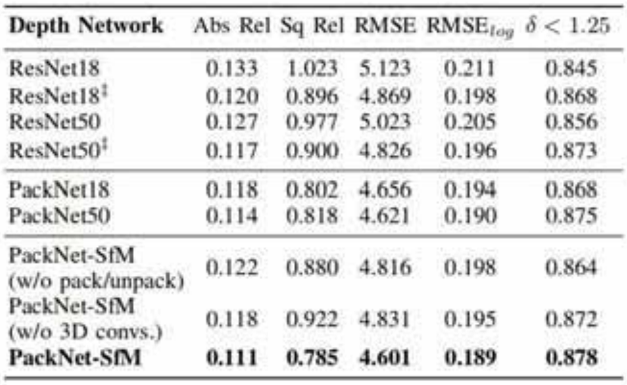
Experimental Results (KITTI)



Experimental Results

Better use of network capacity...

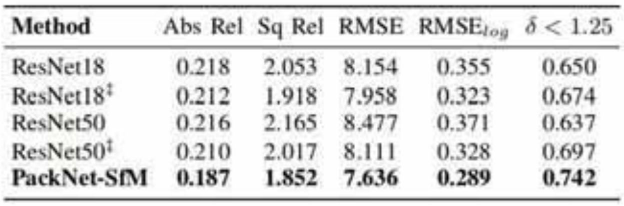




Depth Network	Abs Rel	Sq Rel	RMSE	RMSE$_{log}$	$\delta < 1.25$
ResNet18	0.133	1.023	5.123	0.211	0.845
ResNet18‡	0.120	0.896	4.869	0.198	0.868
ResNet50	0.127	0.977	5.023	0.205	0.856
ResNet50‡	0.117	0.900	4.826	0.196	0.873
PackNet18	0.118	0.802	4.656	0.194	0.868
PackNet50	0.114	0.818	4.621	0.190	0.875
PackNet-SfM (w/o pack/unpack)	0.122	0.880	4.816	0.198	0.864
PackNet-SfM (w/o 3D convs.)	0.118	0.922	4.831	0.195	0.872
PackNet-SfM	**0.111**	**0.785**	**4.601**	**0.189**	**0.878**

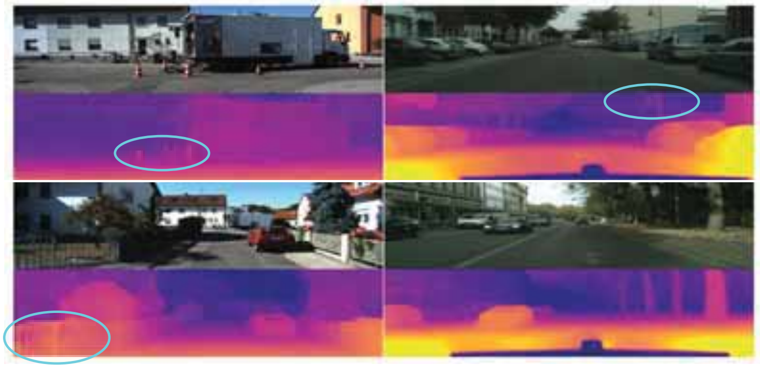
***And** better generalization!*

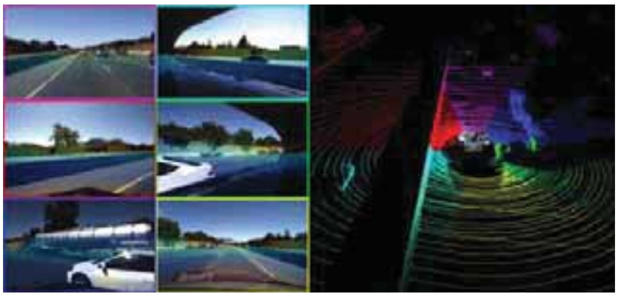
(KITTI → NuScenes)



Method	Abs Rel	Sq Rel	RMSE	RMSE$_{log}$	$\delta < 1.25$
ResNet18	0.218	2.053	8.154	0.355	0.650
ResNet18‡	0.212	1.918	7.958	0.323	0.674
ResNet50	0.216	2.165	8.477	0.371	0.637
ResNet50‡	0.210	2.017	8.111	0.328	0.697
PackNet-SfM	**0.187**	**1.852**	**7.636**	**0.289**	**0.742**

Experimental Results





DDAD: Dense Depth for Autonomous Driving
https://github.com/TRI-ML/DDAD

Frontiers of Monocular 3D Perception @CVPR'20
https://sites.google.com/view/mono3d-workshop

Agenda

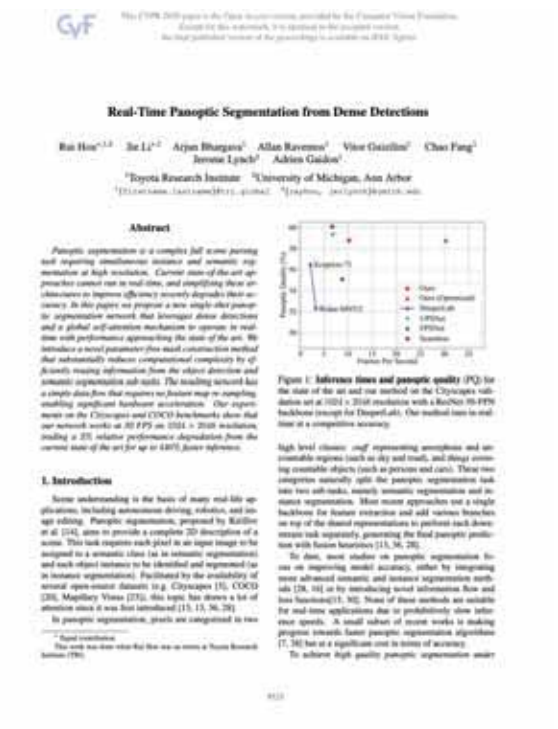
- **Self-Supervised Learning: SuperDepth**
- **Self-Supervised Pseudo-Lidar Networks**
- **Real-time Panoptic Segmentation**

Publication

Real-Time Panoptic Segmentation from Dense Detections

R. Hou, J. Li, A. Bhargava, A. Raventos, F. Guizilini, C. Fang, J. Lynch, A. Gaidon

CVPR 2020, oral presentation



Real-Time Panoptic Segmentation from Dense Detections

Rui Hou*[1,2] Jie Li*[1] Arjun Bhargava[1] Allan Raventos[1] Vitor Guizilini[1] Chao Fang[1]
Jerome Lynch[2] Adrien Gaidon[1]
[1]Toyota Research Institute [2]University of Michigan, Ann Arbor
[1]{firstname.lastname}@tri.global [2]{rayhou, jeelynch}@umich.edu

Abstract

Panoptic segmentation is a complex full scene parsing task requiring simultaneous instance and semantic segmentation at high resolution. Current state-of-the-art approaches cannot run in real-time, and simplifying these architectures to improve efficiency severely degrades their accuracy. In this paper we propose a new single-shot panoptic segmentation network that leverages dense detections and a global self-attention mechanism to operate in real-time with performance approaching the state of the art. We introduce a novel parameter-free mask construction method that substantially reduces computational complexity by efficiently reusing information from the object detection and semantic segmentation sub-tasks. The resulting network has a simple data flow that requires no feature map re-sampling, enabling significant hardware acceleration. Our experiments on the Cityscapes and COCO benchmarks show that our network works at 30 FPS on 1024 × 2048 resolution, trading a 3% relative performance degradation from the current state of the art for up to 440% faster inference.

1. Introduction

Scene understanding is the basis of many real-life applications, including autonomous driving, robotics, and image editing. Panoptic segmentation, proposed by Kirillov et al. [14], aims to provide a complete 2D description of a scene. This task requires each pixel in an input image to be assigned to a semantic class (as in semantic segmentation) and each object instance to be identified and segmented (as in instance segmentation). Facilitated by the availability of several open-source datasets (e.g. Cityscapes [5], COCO [20], Mapillary Vistas [23]), this topic has drawn a lot of attention since it was first introduced [15, 13, 36, 28].

In panoptic segmentation, pixels are categorized in two high level classes: *stuff* representing amorphous and uncountable regions (such as sky and road), and *things* covering countable objects (such as persons and cars). These two categories naturally split the panoptic segmentation task into two sub-tasks, namely semantic segmentation and instance segmentation. Most recent approaches use a single backbone for feature extraction and add various branches on top of the shared representations to perform each downstream task separately, generating the final panoptic prediction with fusion heuristics [13, 36, 28].

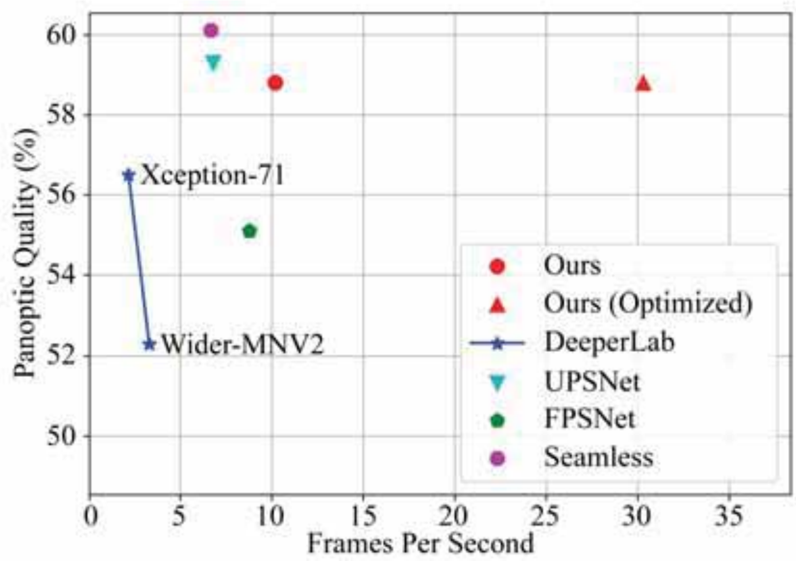
To date, most studies on panoptic segmentation focus on improving model accuracy, either by integrating more advanced semantic and instance segmentation methods [28, 16] or by introducing novel information flow and loss functions[15, 30]. None of these methods are suitable for real-time applications due to prohibitively slow inference speeds. A small subset of recent works is making progress towards faster panoptic segmentation algorithms [7, 38] but at a significant cost in terms of accuracy.

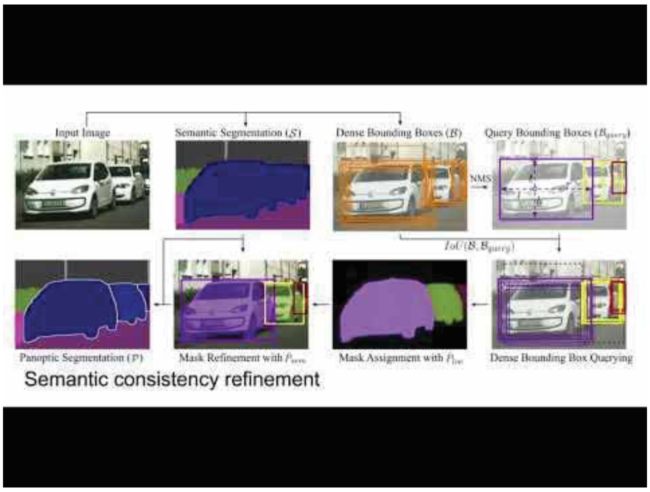
To achieve high quality panoptic segmentation under

Figure 1: **Inference times and panoptic quality** (PQ) for the state of the art and our method on the Cityscapes validation set at 1024 × 2048 resolution with a ResNet 50-FPN backbone (except for DeeperLab). Our method runs in real-time at a competitive accuracy.

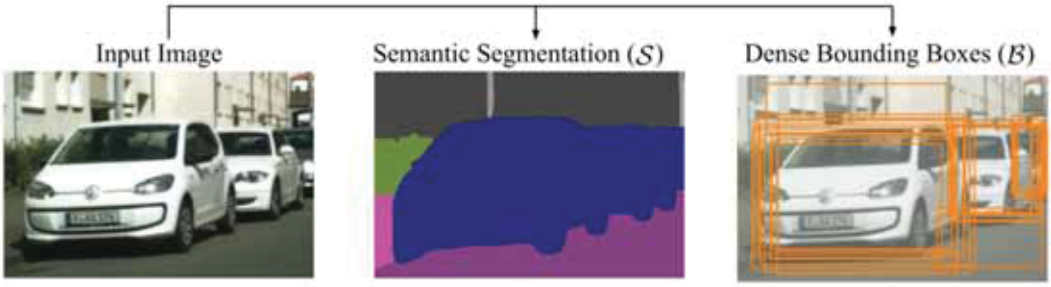
[*] Equal contribution.
This work was done while Rui Hou was an intern at Toyota Research Institute (TRI).

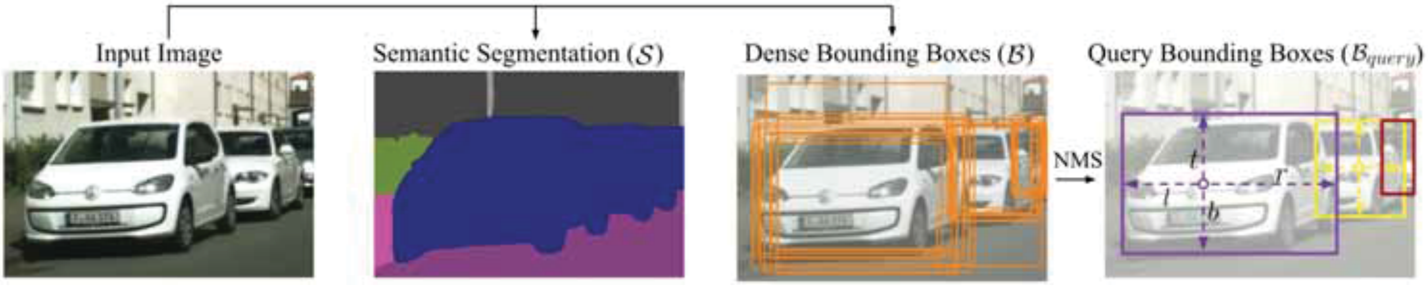
8523

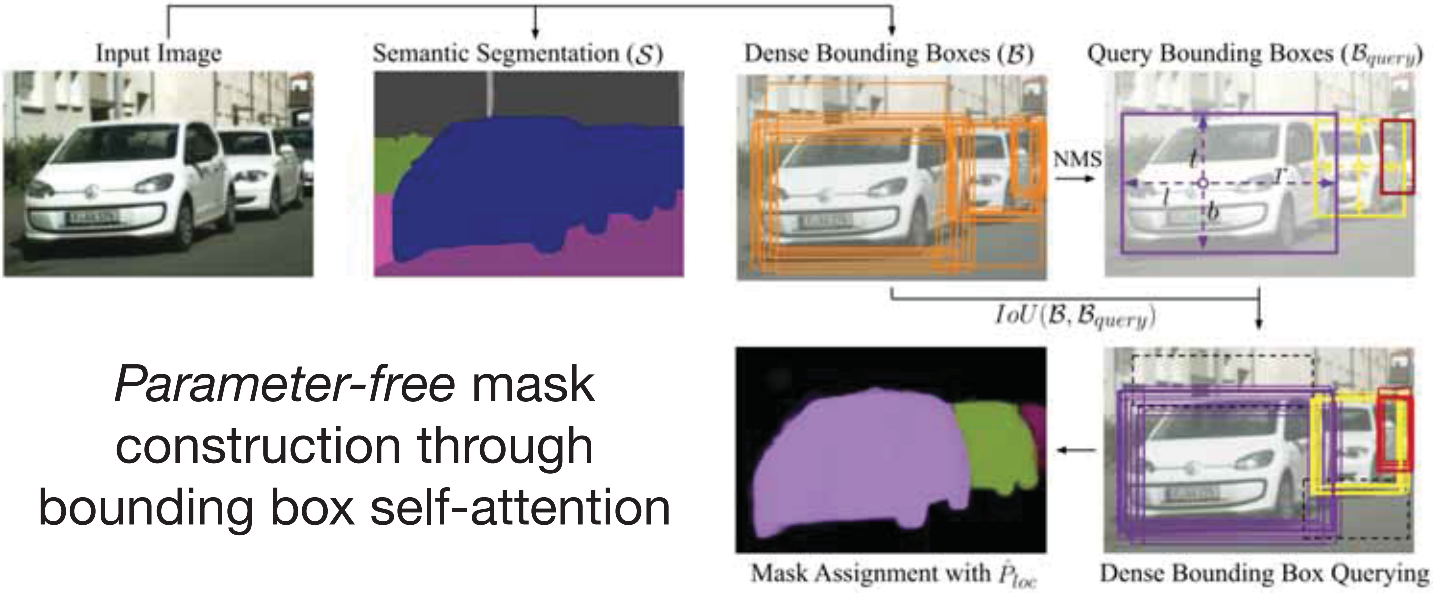



Input Image
Semantic Segmentation (S)
Dense Bounding Boxes (B)
Query Bounding Boxes (B_{query})
NMS
$IoU(B, B_{query})$
Panoptic Segmentation (P)
Mask Refinement with P_{sem}
Mask Assignment with P_{loc}
Dense Bounding Box Querying

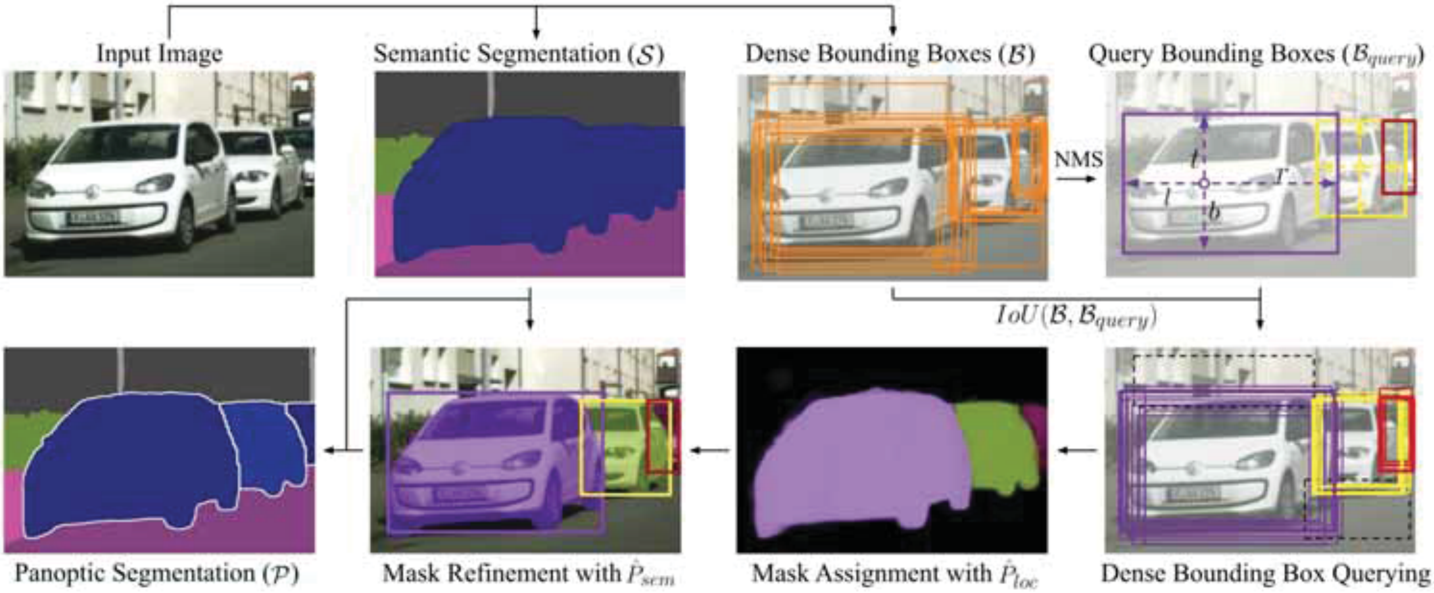
Semantic consistency refinement

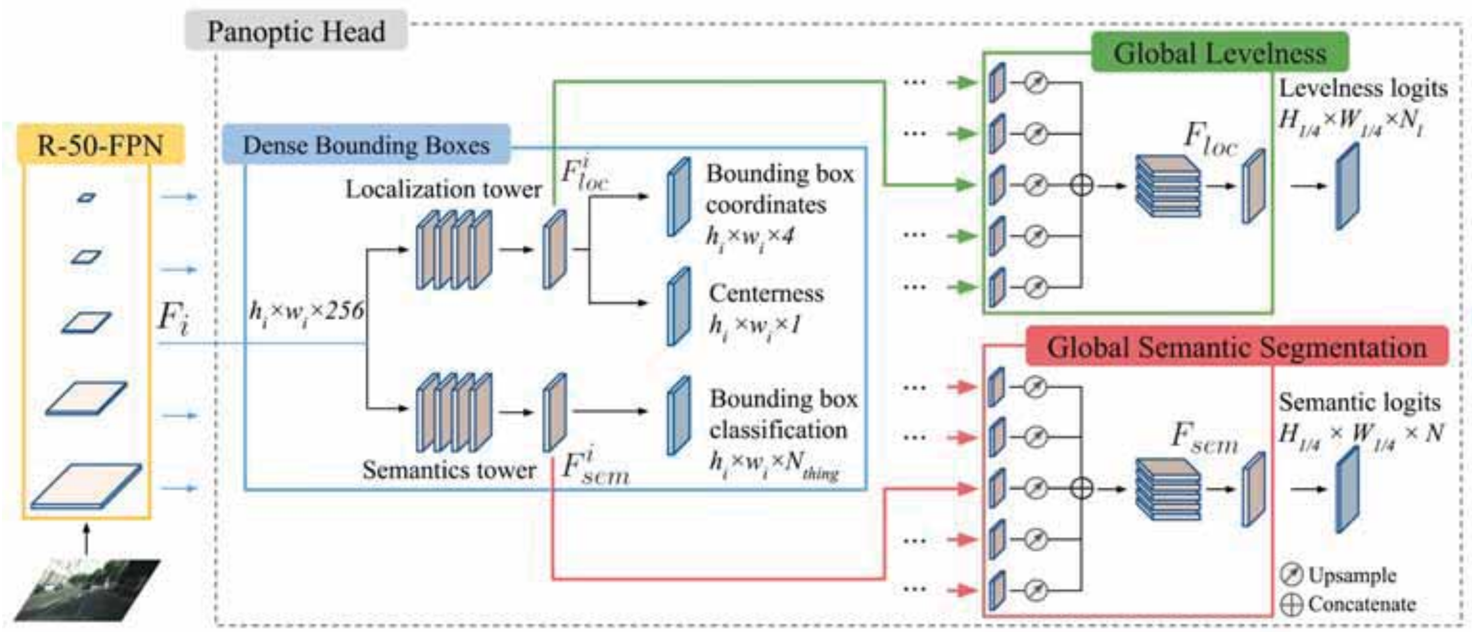

Input Image
Semantic Segmentation ($\mathcal{S}$)
Dense Bounding Boxes ($\mathcal{B}$)


Input Image
Semantic Segmentation ($\mathcal{S}$)
Dense Bounding Boxes ($\mathcal{B}$)
NMS
Query Bounding Boxes ($\mathcal{B}_{query}$)
t
l
r
b

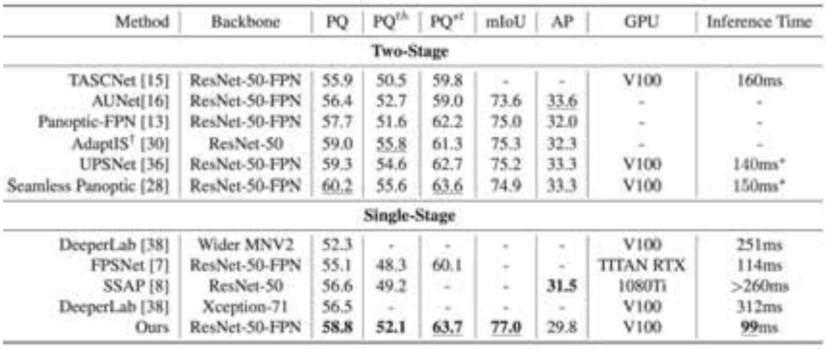

Input Image
Semantic Segmentation ($\mathcal{S}$)
Dense Bounding Boxes ($\mathcal{B}$)
NMS
Query Bounding Boxes ($\mathcal{B}_{query}$)
t
l
r
b
$IoU(\mathcal{B}, \mathcal{B}_{query})$
Mask Assignment with $\hat{P}_{loc}$
Dense Bounding Box Querying

Parameter-free mask construction through bounding box self-attention


Input Image
Semantic Segmentation ($\mathcal{S}$)
Dense Bounding Boxes ($\mathcal{B}$)
NMS
Query Bounding Boxes ($\mathcal{B}_{query}$)
t
l
r
b
$IoU(\mathcal{B}, \mathcal{B}_{query})$
Panoptic Segmentation ($\mathcal{P}$)
Mask Refinement with $\hat{P}_{sem}$
Mask Assignment with $\hat{P}_{loc}$
Dense Bounding Box Querying

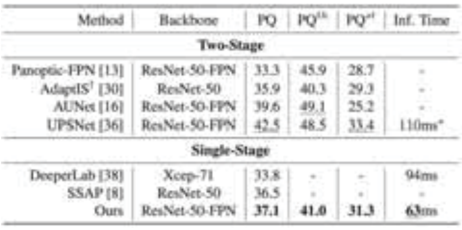

Panoptic Head
R-50-FPN
F_i
Dense Bounding Boxes
Localization tower
$h_i \times w_i \times 256$
F^i_{loc}
Bounding box coordinates
$h_i \times w_i \times 4$
Centerness
$h_i \times w_i \times 1$
Semantics tower
F^i_{sem}
Bounding box classification
$h_i \times w_i \times N_{thing}$
Global Levelness
F_{loc}
Levelness logits
$H_{1/4} \times W_{1/4} \times N_l$
Global Semantic Segmentation
F_{sem}
Semantic logits
$H_{1/4} \times W_{1/4} \times N$
Upsample
Concatenate

Cityscapes (val)

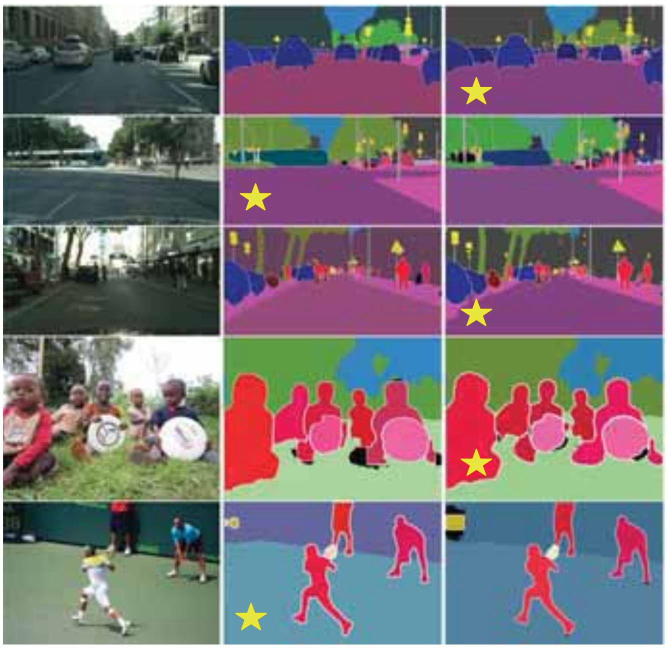


Method	Backbone	PQ	PQ^{th}	PQ^{st}	mIoU	AP	GPU	Inference Time
Two-Stage								
TASCNet [15]	ResNet-50-FPN	55.9	50.5	59.8	-	-	V100	160ms
AUNet[16]	ResNet-50-FPN	56.4	52.7	59.0	73.6	33.6	-	-
Panoptic-FPN [13]	ResNet-50-FPN	57.7	51.6	62.2	75.0	32.0	-	-
AdaptIS† [30]	ResNet-50	59.0	55.8	61.3	75.3	32.3	-	-
UPSNet [36]	ResNet-50-FPN	59.3	54.6	62.7	75.2	33.3	V100	140ms*
Seamless Panoptic [28]	ResNet-50-FPN	60.2	55.6	63.6	74.9	33.3	V100	150ms*
Single-Stage								
DeeperLab [38]	Wider MNV2	52.3	-	-	-	-	V100	251ms
FPSNet [7]	ResNet-50-FPN	55.1	48.3	60.1	-	-	TITAN RTX	114ms
SSAP [8]	ResNet-50	56.6	49.2	-	-	**31.5**	1080Ti	>260ms
DeeperLab [38]	Xception-71	56.5	-	-	-	-	V100	312ms
Ours	ResNet-50-FPN	**58.8**	**52.1**	63.7	77.0	29.8	V100	99ms

COCO (val)

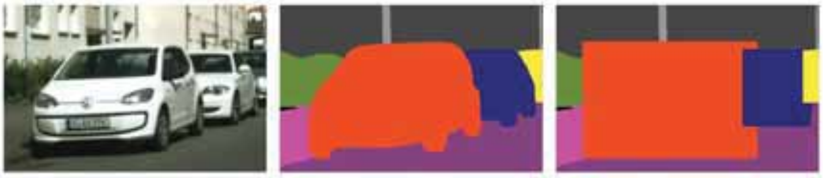


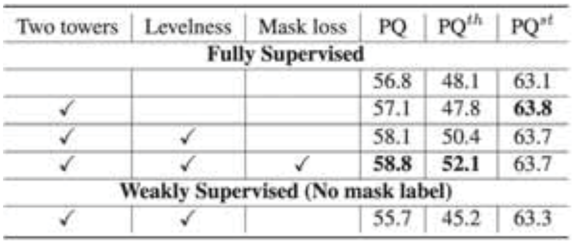
Method	Backbone	PQ	PQ^{th}	PQ^{st}	Inf. Time
Two-Stage					
Panoptic-FPN [13]	ResNet-50-FPN	33.3	45.9	28.7	-
AdaptIS† [30]	ResNet-50	35.9	40.3	29.3	-
AUNet [16]	ResNet-50-FPN	39.6	49.1	25.2	-
UPSNet [36]	ResNet-50-FPN	42.5	48.5	33.4	110ms*
Single-Stage					
DeeperLab [38]	Xcep-71	33.8	-	-	94ms
SSAP [8]	ResNet-50	36.5	-	-	-
Ours	ResNet-50-FPN	**37.1**	**41.0**	**31.3**	**63**ms



☆ Ours

Supervision: Weak = 95% Strong

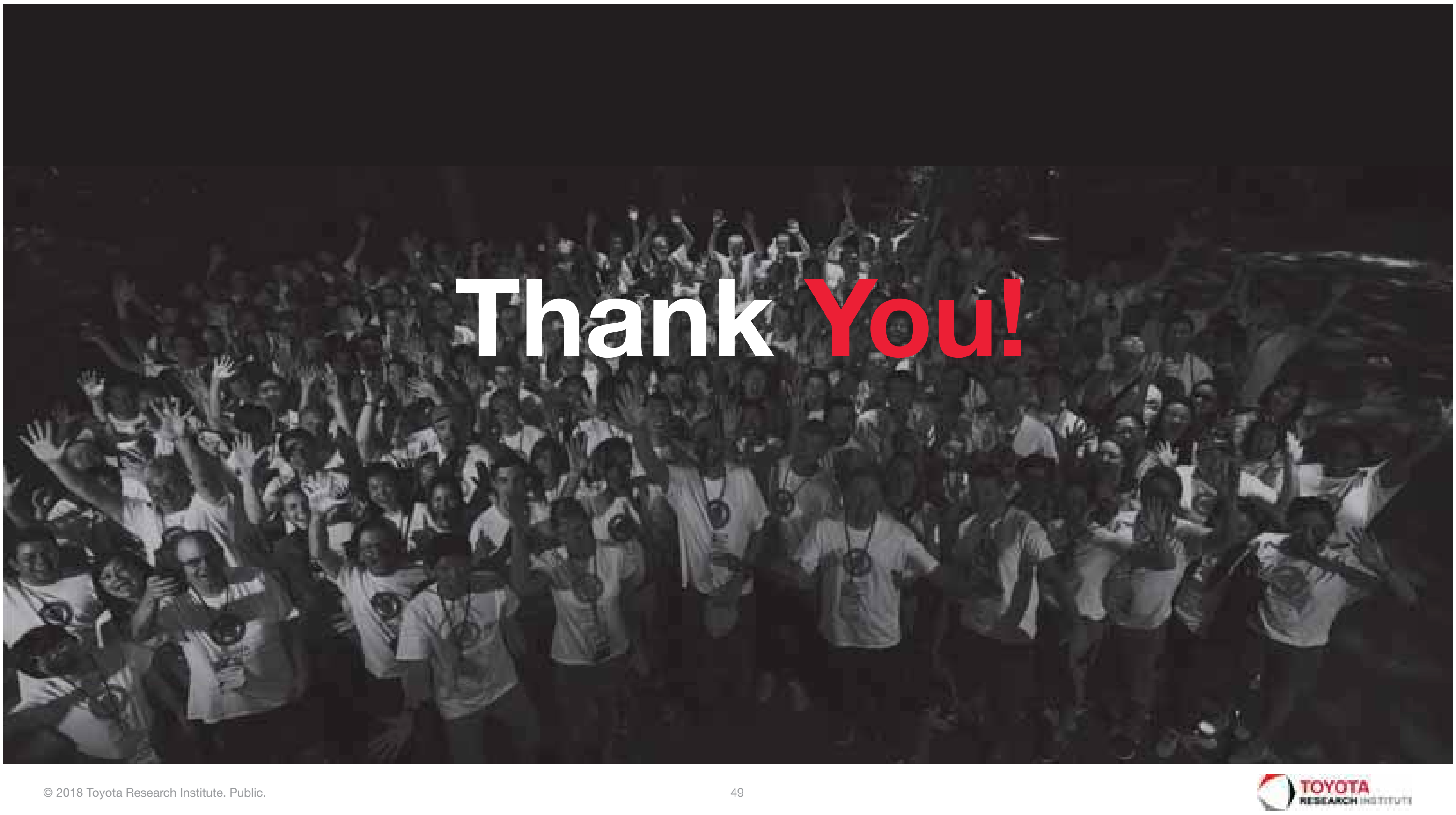




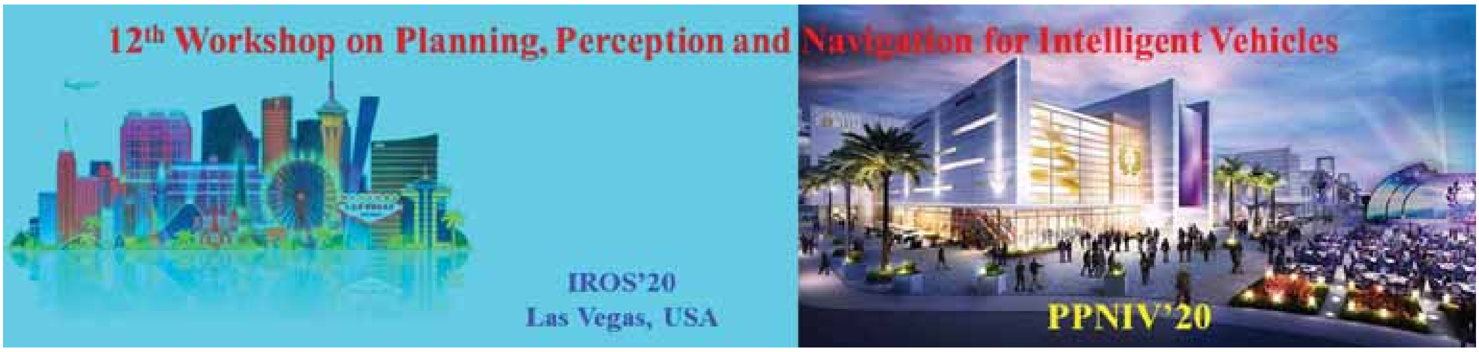
Two towers	Levelness	Mask loss	PQ	PQth	PQst
		Fully Supervised			
			56.8	48.1	63.1
✓			57.1	47.8	**63.8**
✓	✓		58.1	50.4	63.7
✓	✓	✓	**58.8**	**52.1**	63.7
		Weakly Supervised (No mask label)			
✓	✓		55.7	45.2	63.3

Conclusion

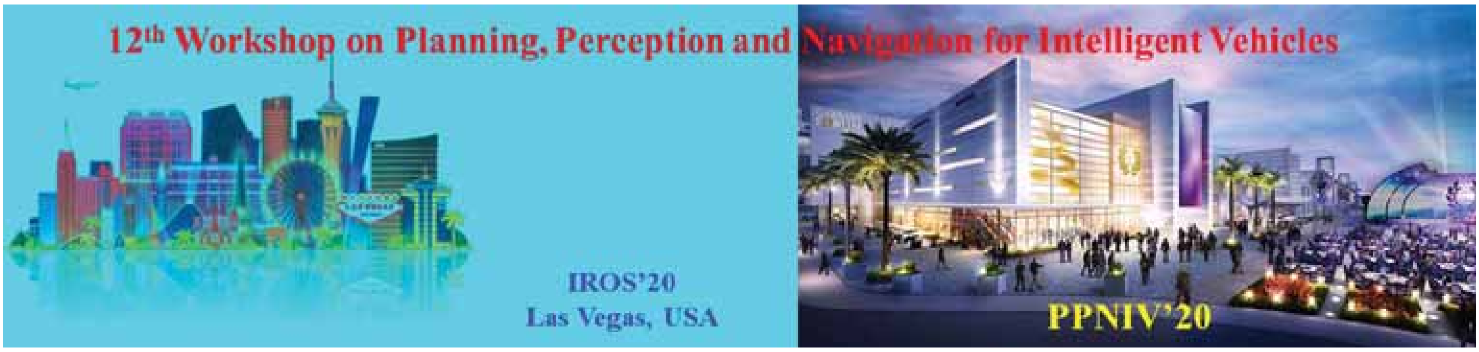
- Building truly autonomous cars requires machine learning
- The supervised learning approach does not scale
- We need to go beyond supervised learning and be able to learn from structured, unlabeled data



Thank You!


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

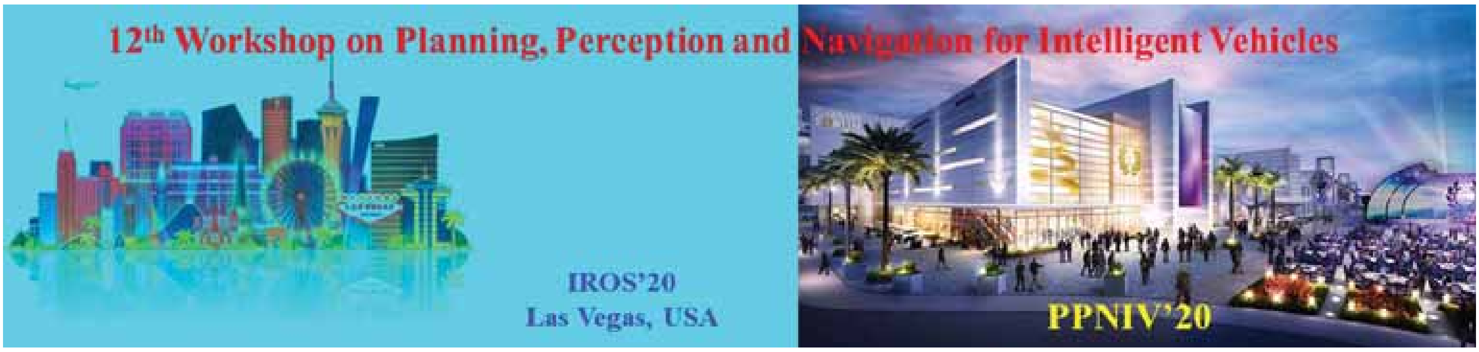
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Keynote speaker: **Daniela Rus**
(MIT, USA)

Understanding Risk and Social Behavior Improves Decision Making for Autonomous Vehicles

Abstract: Deployment of autonomous vehicles on public roads promises increases in efficiency and safety, and requires evaluating risk, understanding the intent of human drivers, and adapting to different driving styles. Autonomous vehicles must also behave in safe and predictable ways without requiring explicit communication. This talk describes how to integrate risk and behavior analysis in the control look of an autonomous car. I will describe how Social Value Orientation (SVO), which captures how an agent's social preferences and cooperation affect their interactions with others by quantifying the degree of selfishness or altruism, can be integrsted in decision making and provide recent examples of developing and deploying self-driving vehicles with adaptation capabilities.

Biograpghy: Daniela Rus is the Andrew (1956) and Erna Viterbi Professor of Electrical Engineering and Computer Science, Director of the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT, and Deputy Dean of Research in the Schwarzman College of Computing at MIT. She is also a visiting fellow at Mitre Corporation. Rus's research interests are in robotics and artificial intelligence. The key focus of her research is to develop the science and engineering of autonomy. Rus is a Class of 2002 MacArthur Fellow, a fellow of ACM, AAAI and IEEE, and a member of the National Academy of Engineering and of the American Academy of Arts and Sciences. She is the recipient of the Engelberger Award for robotics. She earned her PhD in Computer Science from Cornell University.


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Risk and Social Behavior for Decision Making for Autonomous Vehicles

Daniela Rus

(with Alexander Amini, Igor Gilitschenski, Teddy Ort, Alyssa Pierson, Wilko Schwarting, Sertac Karaman)

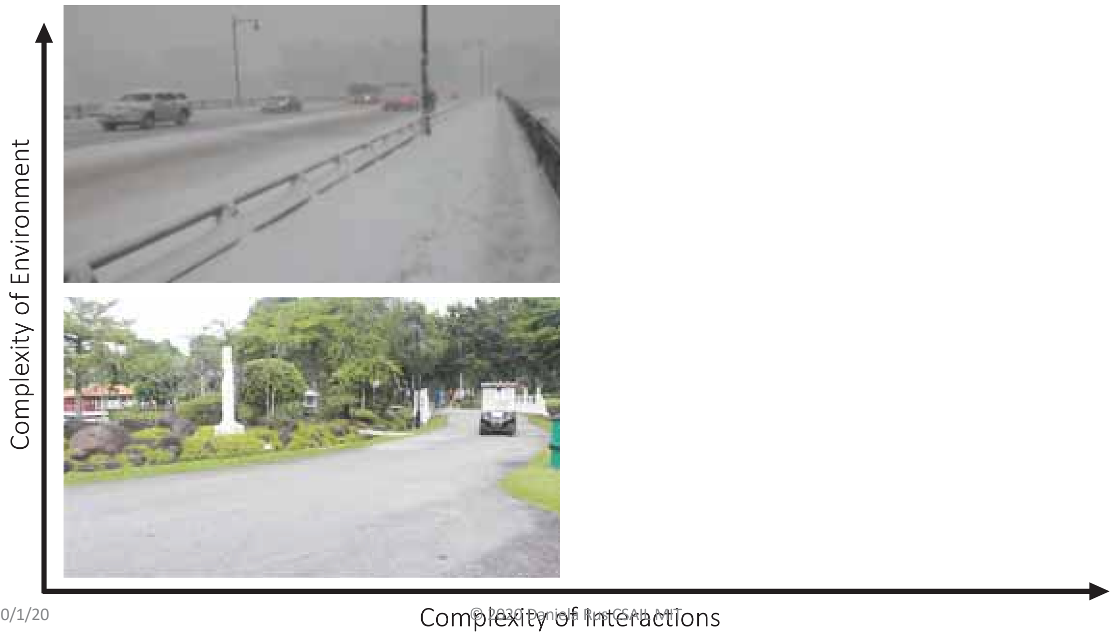
CSAIL, MIT

SMART, MIT

Autonomous Vehicles: Where are they?

Complexity of Environment
Complexity of Interactions

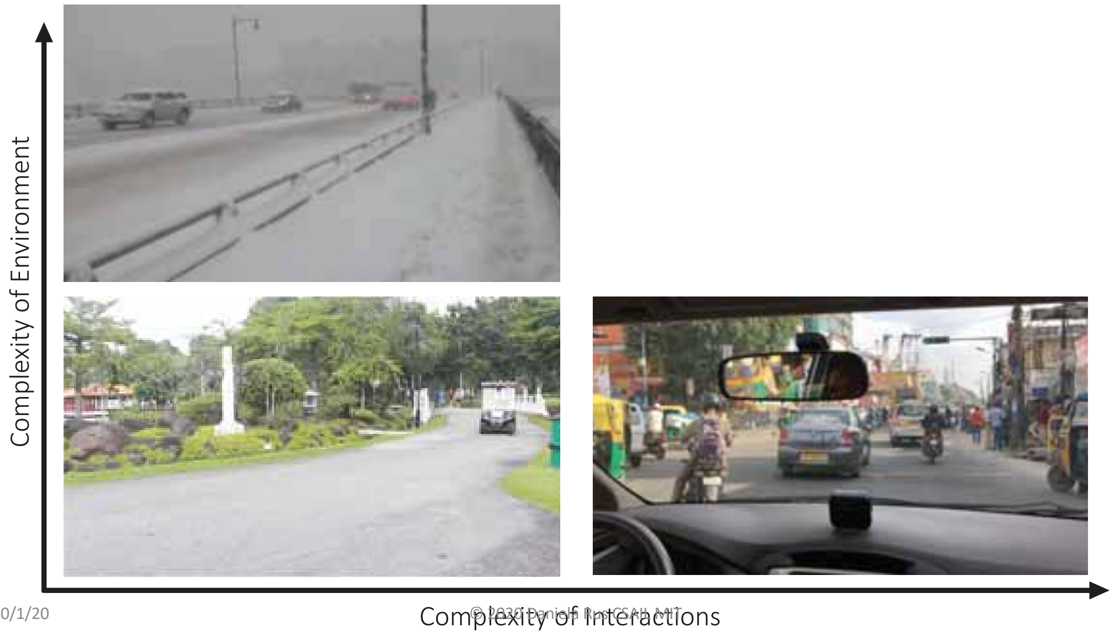
Autonomous Vehicles: Where are they?



10/1/20

© 2020 Daniela Rus CSAIL MIT

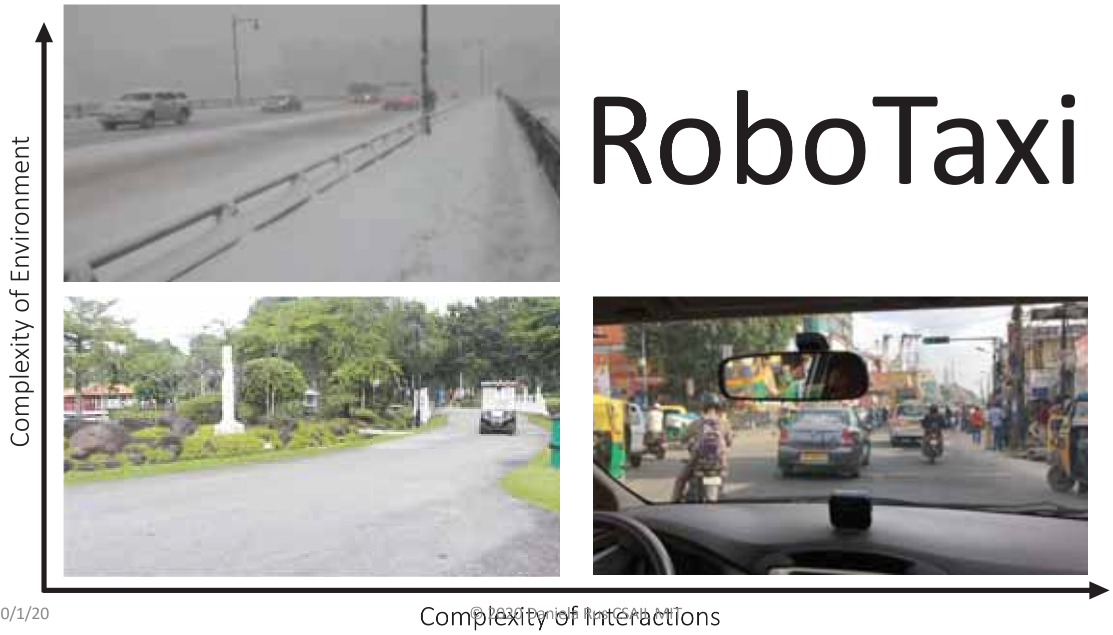
Autonomous Vehicles: Where are they?



10/1/20

© 2020 Daniela Rus CSAIL MIT

Autonomous Vehicles: Where are they?



RoboTaxi

Outline

How can we enable self-driving vehicles to operate in more complex environments?

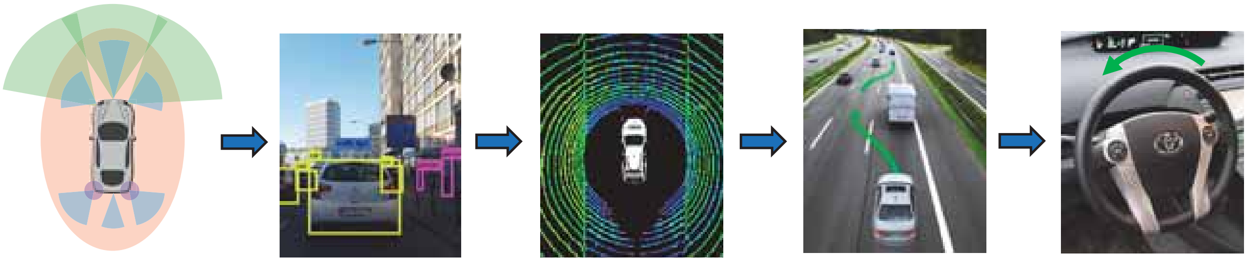
How can we incorporate risk in the control loop?

How can we handle congestion and interactions with human-driven cars?

Increased Capabilities: Learning to Drive from Humans

Classical autonomous driving pipeline

Separate problem into smaller sub-modules, tackle each independently



Sensor Fusion
What's happening around me?

Detection
Where are obstacles?

Localization
Where am I relative to the obstacles?

Planning
Where do I go?

Actuation
What control signals to take?

Learning our autonomous controller

Autonomous systems need the ability to handle a wide range of scenarios using raw and complex perception sensors

ight-time riving

o Lane arkings

ainy eather

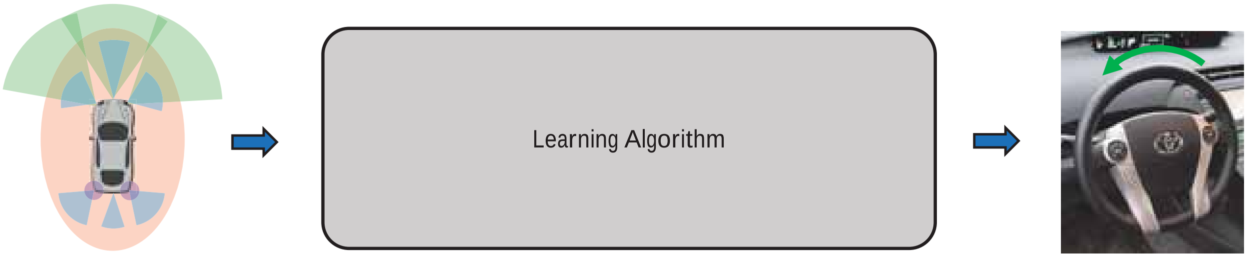
Leveraging large datasets, we learn an underlying representation of driving based on how humans drive in similar situations

10/1/20

© 2020 Daniela Rus CSAIL MIT

End-to-End Learning

Learn the control directly from raw sensor data



Sensor Fusion
What's happening around me?

Learned Model
Underlying representation of how humans drive

Actuation
What control signals to take?

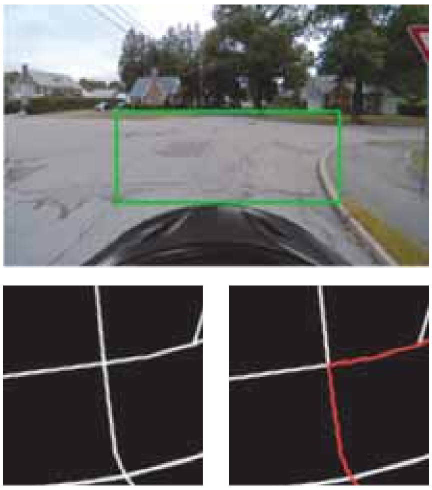
10/1/20

© 2020 Daniela Rus CSAIL MIT

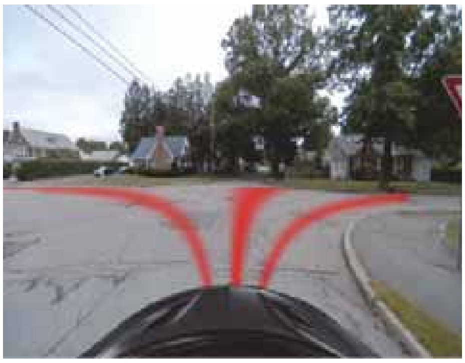
Learning to navigate

a Perce tion
I
ex camera

oarse Ma s
M
ex S



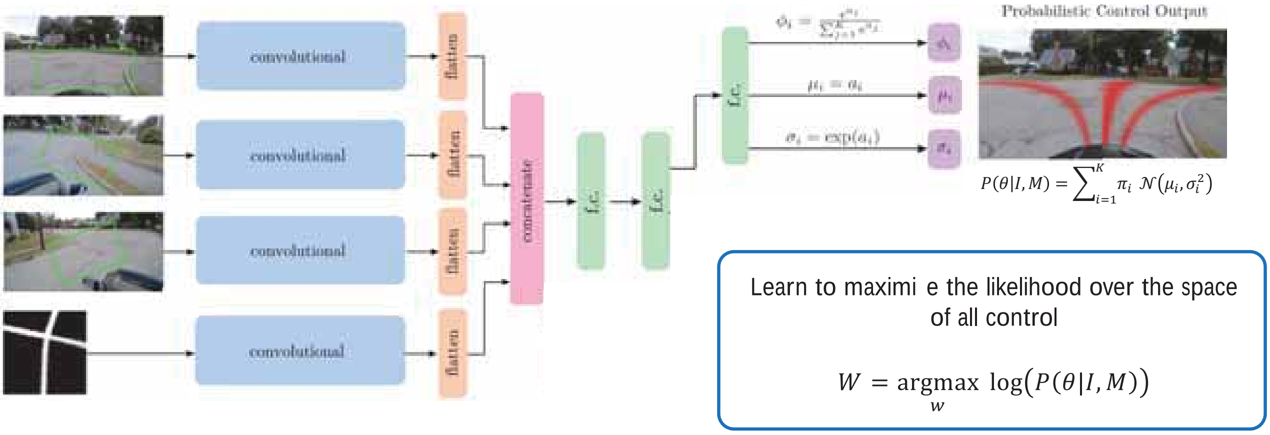
robabilistic ontrol
$P(\theta|I,M)$



$$P(\theta|I,M) = \sum_{i=1}^{K} \pi_i \, \mathcal{N}(\mu_i, \sigma_i^2)$$

End-to-end optimization formulation

Learn a continuous probability distribution over the space of all control $P(\theta|I,M)$

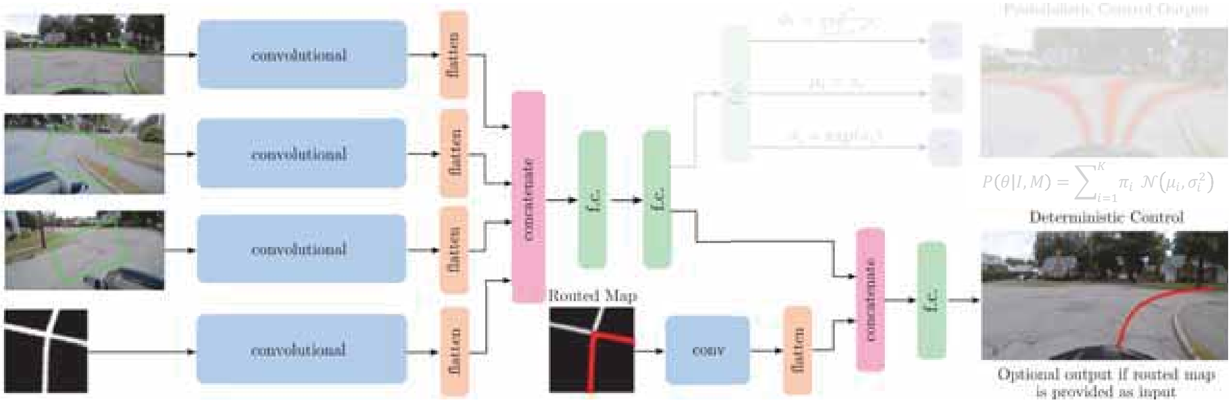


Learn to maximi e the likelihood over the space of all control

$$W = \underset{w}{\operatorname{argmax}} \log(P(\theta|I,M))$$

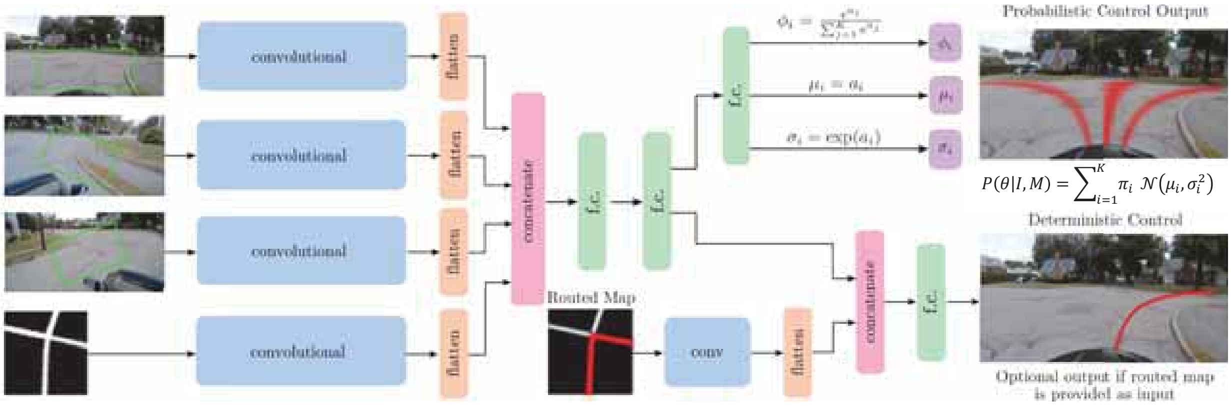
End-to-end optimization formulation

nput a route to compute a deterministi ontrol ommand for navigation

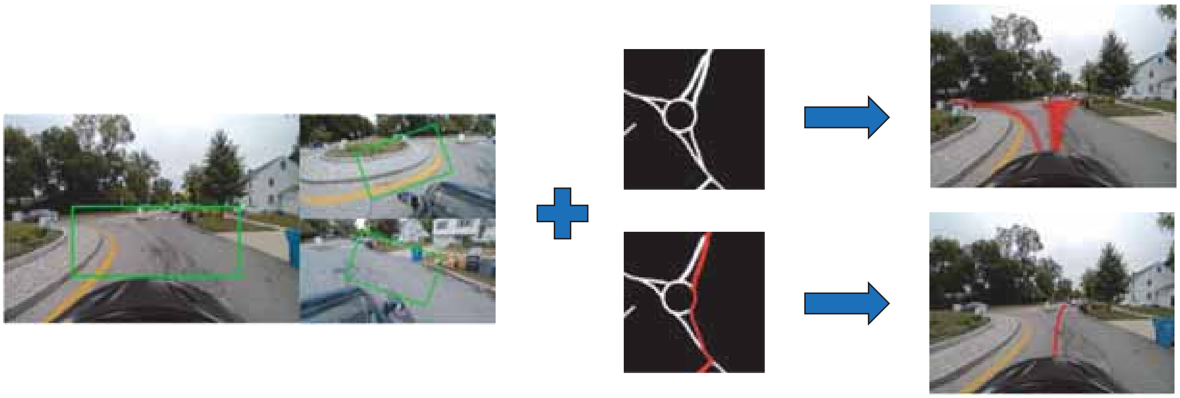


End-to-end optimization formulation

ntire model is trained end-to-end ithout an human la ellin or annotations



Generalization to new roads



Perce tion n ut

oarse Street a

ontrol ut ut

odel learned to generali e to ne roads and even ne t es o interse tions ex roundabouts never included in training

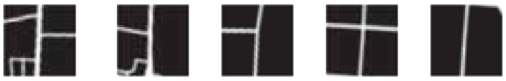
Correcting pose based on visual perception

hat do we do when our S is not accurate or even not available $\Rightarrow Var[P(M)] \gg 0$

$$P(M|I) = \mathbb{E}_{\theta}\left[\frac{P(\theta|I,M)}{\mathbb{E}_{M'}P(\theta|I,M')}P(M)\right]$$

iven this image of your surroundings

which map are you most likely in

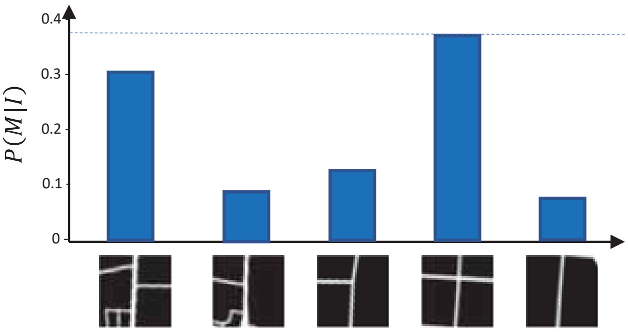


Correcting pose based on visual perception

hat do we do when our S is not accurate or even not available $\Rightarrow Var[P(M)] \gg 0$

$$P(M|I) = \mathbb{E}_{\theta}\left[\frac{P(\theta|I,M)}{\mathbb{E}_{M'}P(\theta|I,M')}P(M)\right]$$

iven this image of your surroundings
which map are you in

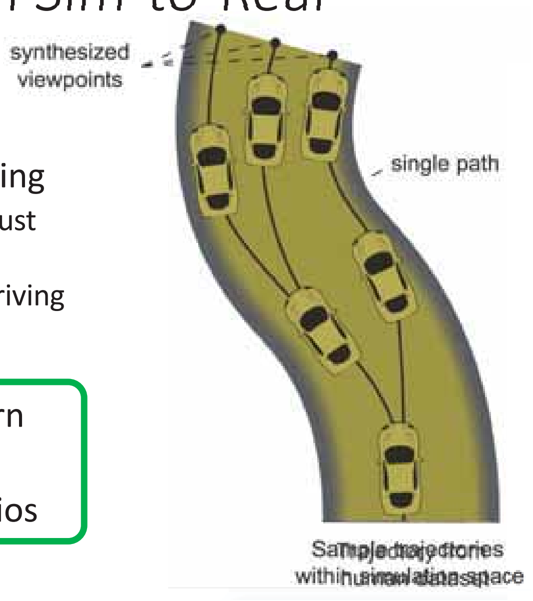


10/1/20

Increasing Scope of Training with Sim-to-Real

- End-to-end perception-to-control learning
- Imitate human driving through supervised learning
  - Dangerous to collect data from situations vehicles must be able to handle
  - Requires large amounts of “gold-standard” human driving
  - Difficulty in transferring to new domains, edge cases

- Allow agents to autonomously navigate and learn how to drive **without human supervision**
- Real world edge-cases and safety-critical scenarios


synthesized viewpoints
single path
Sample trajectories within simulation space
Trajectories in human data space

Related Works

Model-based Simulation

- Lacks photorealism
- Does not capture semantic complexity
- Does not transfer to real world (current state of art)
- [CARLA] Dosovitskiy et al (2017)
- [Torcs] Wymann et al (2000)

Domain Transfer

- Possible to transfer to real world
- Transfer limited to textures
- Lacks photorealism and semantic complexity
- [Wayve] Bewley et al (2018)
- Pan et al (2017)

Data-Driven Simulation

- Photorealistic + transferable
- Not scalable to large scale driving environments
  - [Gibson] Xia et al (2018)
- Deformities from non-realistic assumptions
  - [NVIDIA] Bojarski et al (2016)

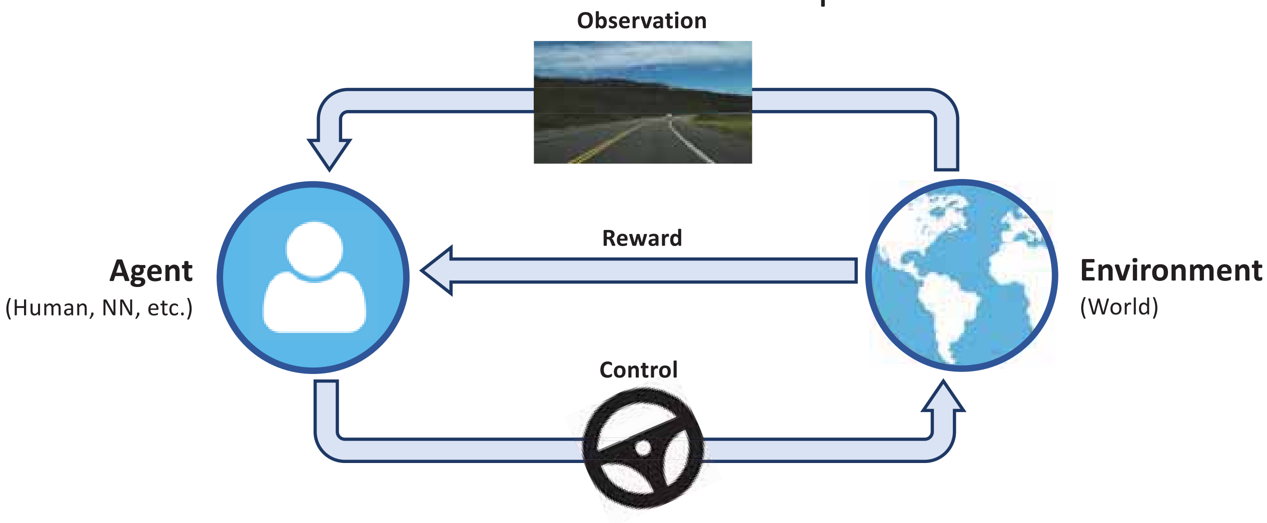
Approach

1. **Photorealistic data-driven simulation** engine for synthesizing new control trajectories.

2. **Real-world transferable reinforcement learning**. End-to-end without human imitation.

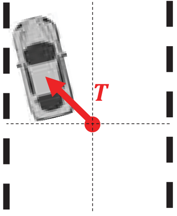
10/1/20

End-to-end without human supervision



10/1/20

Optimizing high level reward functions



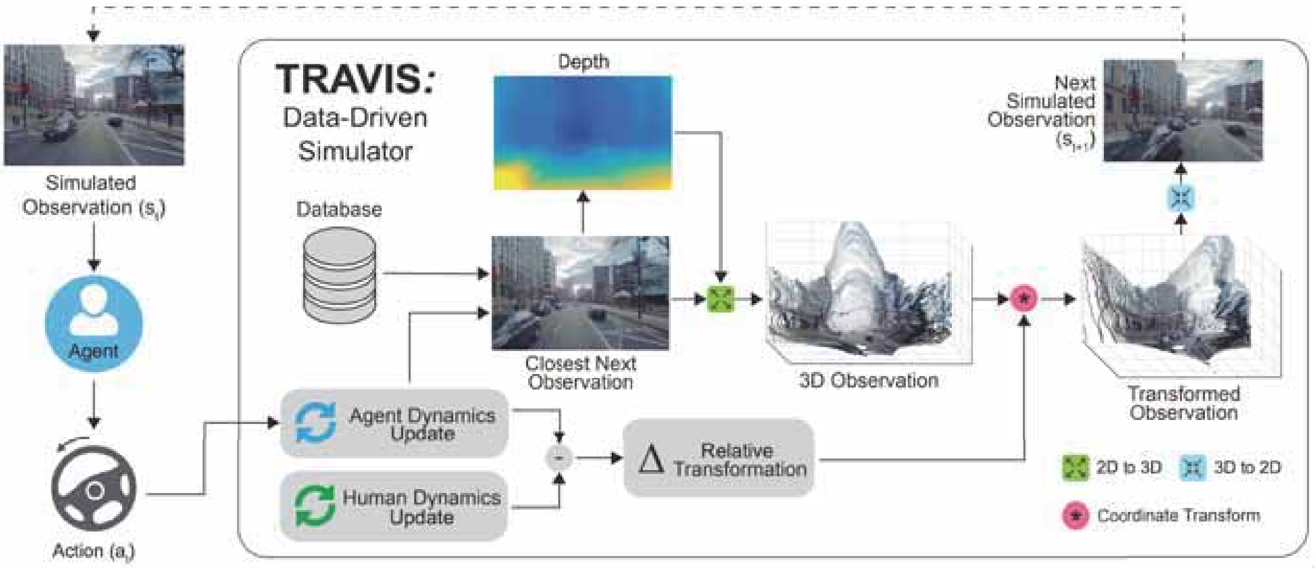
$$r_t = \begin{cases} 1 & \text{if } \|T\| < \varepsilon \\ 0 & \text{otherwise} \implies \text{crash} \end{cases}$$

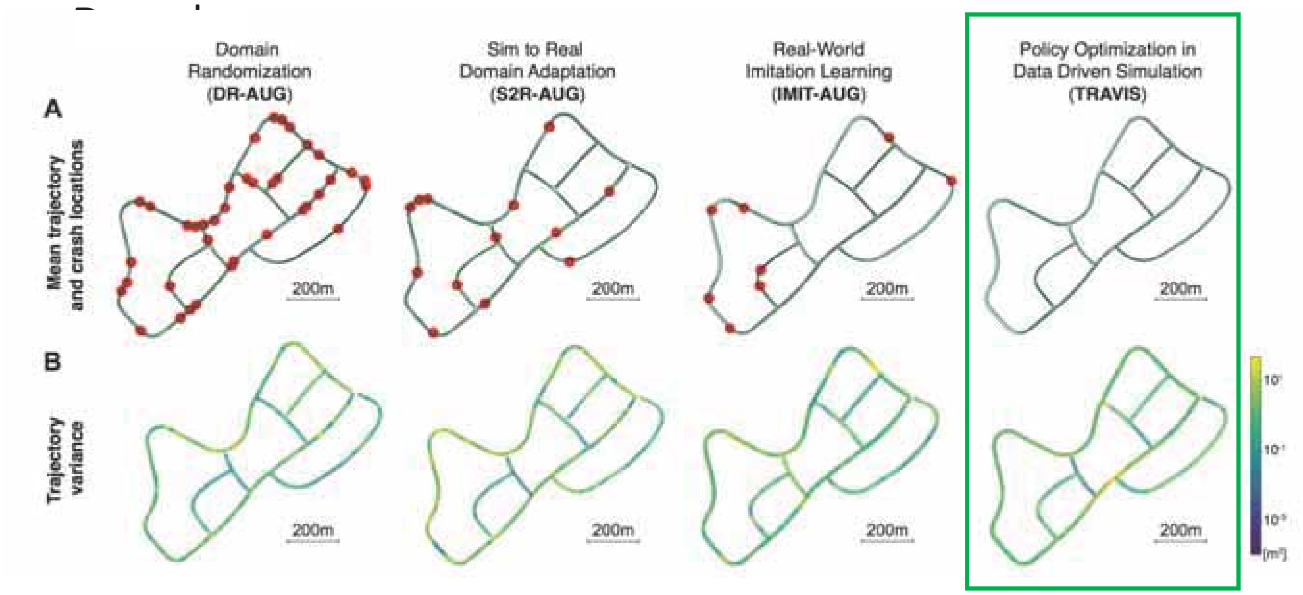
Instead of imitating a human driver, directly **optimize the agent to maximize its own rewards**

$$\max_{\pi_\theta} \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_t \gamma^t r_t\right]$$

$$\tau = \{(a_1, s_1, r_1), (a_2, s_2, r_2), \dots\} \sim \pi_\theta(a_t|s_t)$$

End-to-end without human supervision




Domain Randomization (DR-AUG)
Sim to Real Domain Adaptation (S2R-AUG)
Real-World Imitation Learning (IMIT-AUG)
Policy Optimization in Data Driven Simulation (TRAVIS)
A
Mean trajectory and crash locations
B
Trajectory variance
200m
10¹
10⁻¹
10⁻³
[m²]

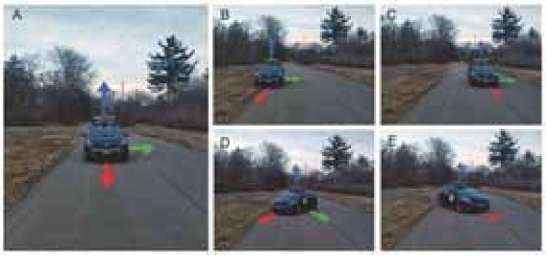
10/1/20

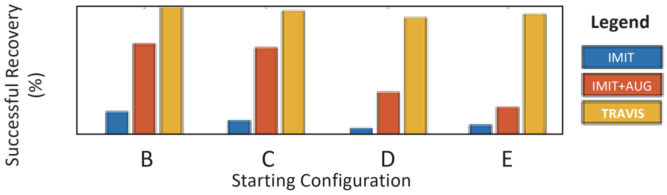
© 2020 Daniela Rus CSAIL MIT

Results

Direct deployment to real-world without any adjustments

Superior robustness to recover from challenging off-orientations



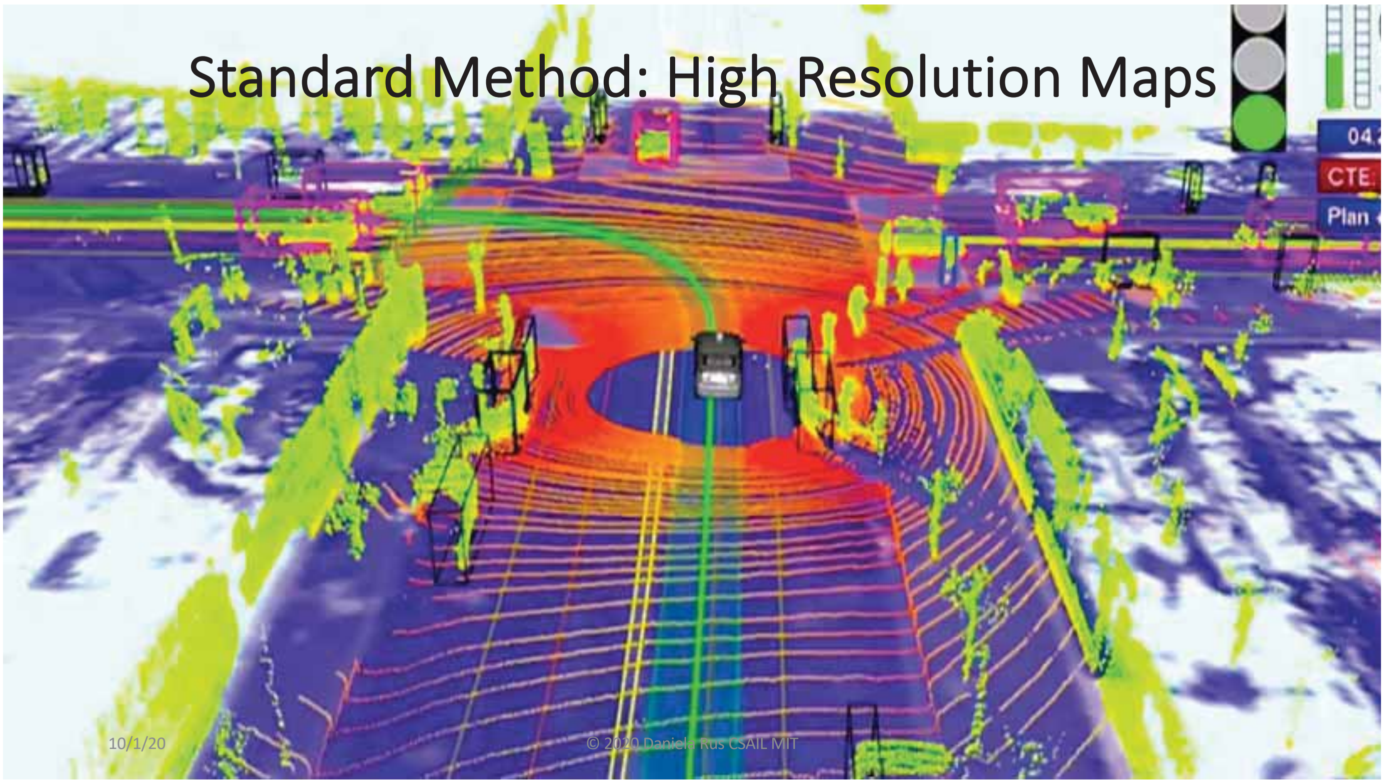


10/1/20

© 2020 Daniela Rus CSAIL MIT

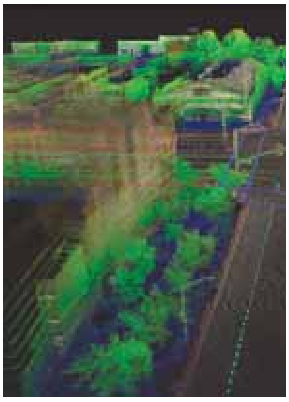
Challenging Environments: No Maps, Weather

Standard Method: High Resolution Maps

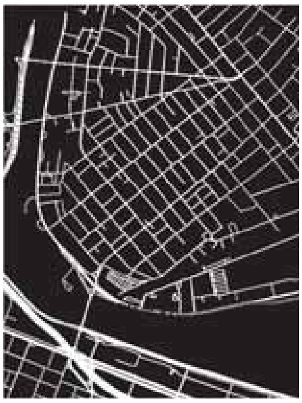


Challenges with Maps:

- Scalability
  - Maps can grow large; hard to store or transmit regions larger than small cities
- Maintenance
  - Maps must be maintained, small changes in the world can cause localization failure
- Coverage
  - Rural areas not densely populated and the landscape can change rapidly
- Features



San Francisco, 4 TB
[Puttagunta, Civil Maps]

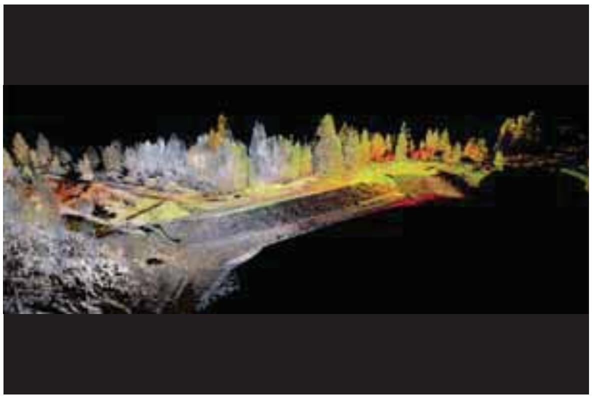


Earth, 40 GB
[Planet.osm]

10/1/20 © 2020 Daniela Rus CSAIL MIT

10/1/20 © 2020 Daniela Rus CSAIL MIT

Vision and LiDAR



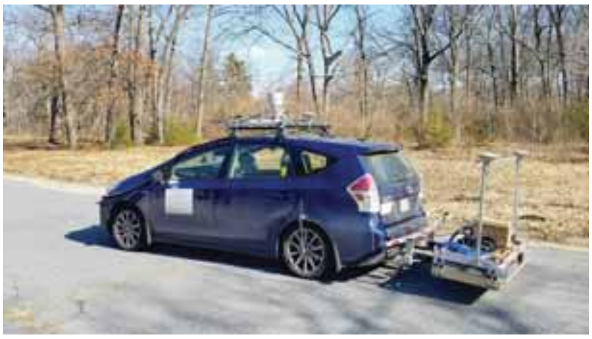
10/1/20 © 2020 Daniela Rus CSAIL MIT

Changing Environments are Challenging

10/1/20 © 2020 Daniela Rus CSAIL MIT

Localizing Ground Penetrating Radar

- Use Ground Penetrating Radar to build a map of underground features
- Soil content, type, layers can be reliably detected down to 2-3m
- Radar is unaffected by surface parameters like light and lidar



10/1/20

Related Work

Radar-based Perception

- With LiDAR
  [Rasshofer, ARS 2005]
- With Cameras
  [Mori, IV 2007]
- For SLAM
  [Schuster, Ward, IV 2016]

Appearance Modeling

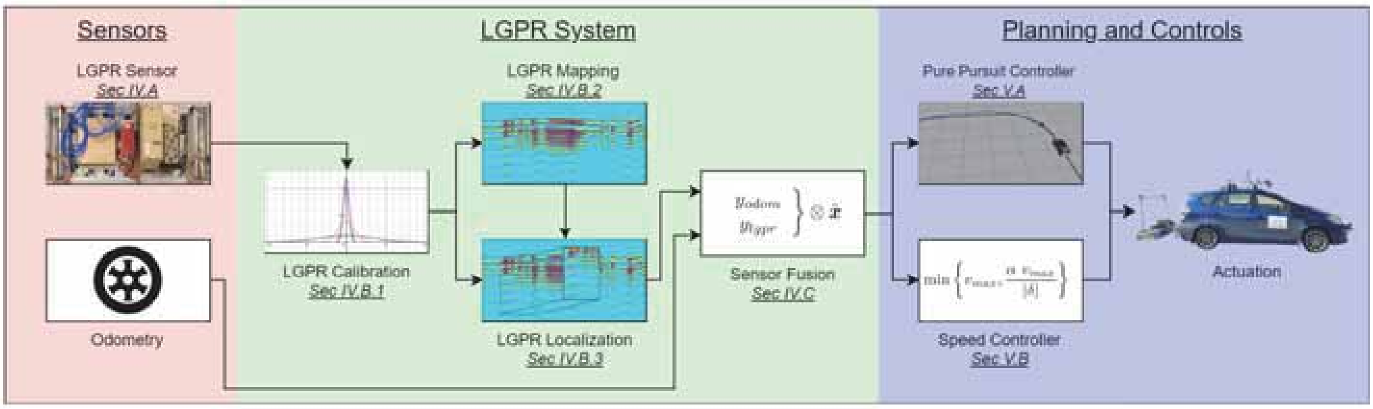
- Dynamic Object Removal
  [Bescos, RA-L 2018]
- Stable Features
  [Dymczyk, 3DV 2007]
- Landmark Selection
  [Burki, JFR 2019]

Ground Penetrating Radar

- Soil Analysis
  [Rea, Water Resources Research 1998]
- Autonomous Analysis
  [Williams, IGARSS 2012]
- Localization
  [Cornick, JFR 2016]

10/1/20

LGPR System



Sensors

LGPR Sensor
Sec IV.A

Odometry

LGPR System

LGPR Calibration
Sec IV.B.1

LGPR Mapping
Sec IV.B.2

LGPR Localization
Sec IV.B.3

$\left.\begin{matrix} y_{odom} \\ y_{lgpr} \end{matrix}\right\} \otimes \hat{x}$

Sensor Fusion
Sec IV.C

Planning and Controls

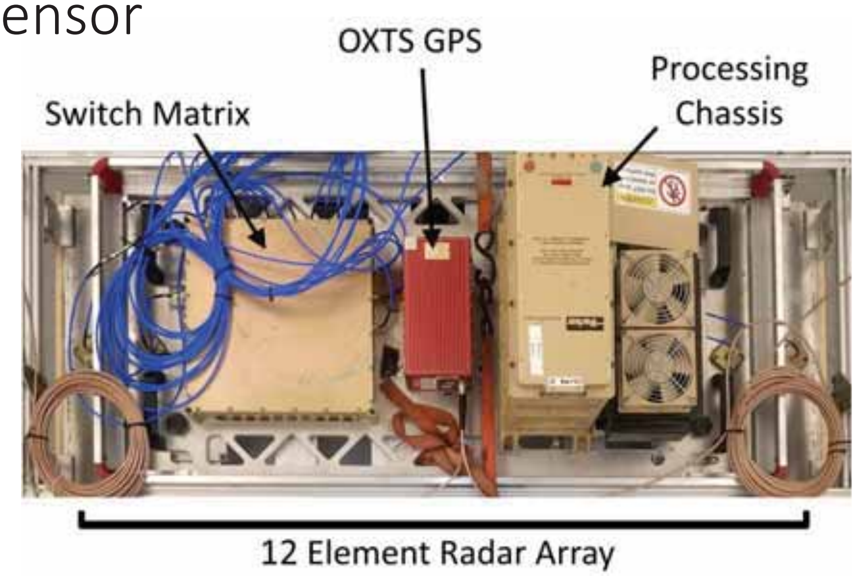
Pure Pursuit Controller
Sec V.A

$\min\left\{ v_{max}, \frac{\alpha \, v_{max}}{|\delta|} \right\}$

Speed Controller
Sec V.B

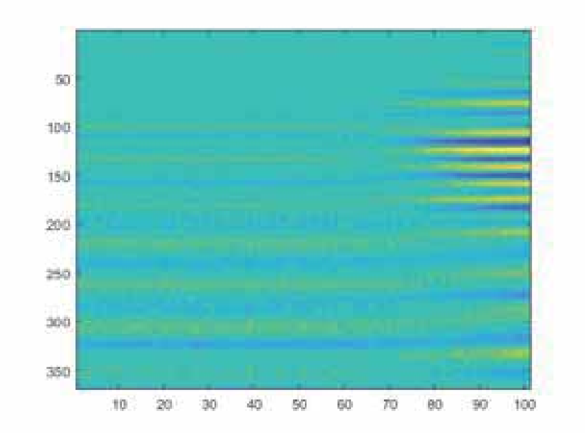
Actuation

LGPR Sensor



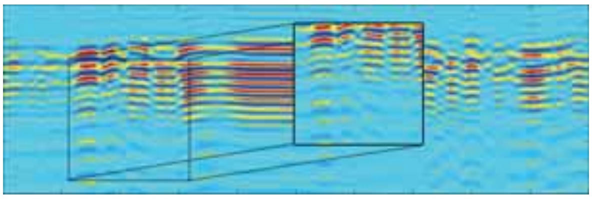
Mapping

- Sensor records 2D scans beneath the vehicle [11x369]
- Data rate is up to 126Hz
- Each scan is stored with a GPS location for localization



Localization

- A single scan is located in the map (5 DOF)
- GPS rough position limits search space
- Interpolation is used between map scans



Correlation:

$$r_{A,B} = \frac{\sum_{i,d} A_{i,d} B_{i,d}}{\sqrt{\sum_{i,d} A_{i,d}^2 B_{i,d}^2}}$$

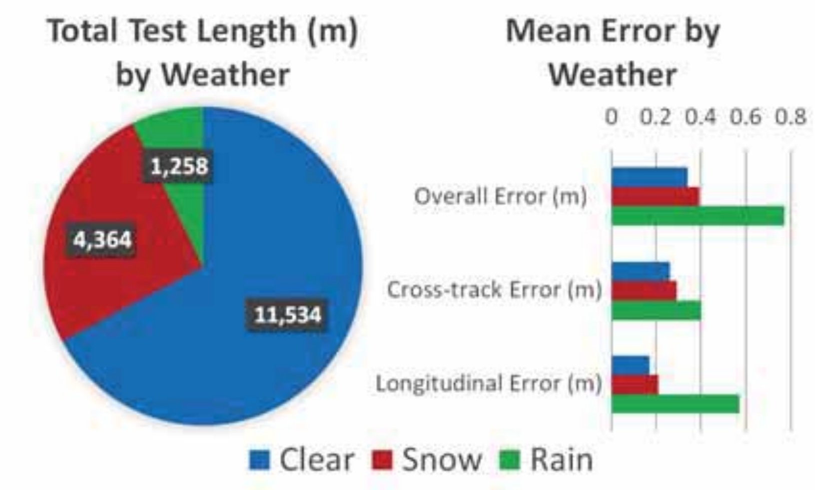
System Evaluation

- Full system implemented on autonomous Prius
- Real-Time-Kinematic GPS Inertial Navigation System for ground-truth

Relative mean error:

$$\frac{1}{n}\sum_{n}\left\|\left(T^{test}_{GNSS,i}-T^{map}_{GNSS,i}\right)-\left(T^{test}_{LGPR,i}-T^{map}_{LGPR,i}\right)\right\|$$

Driving in Weather Results



10/1/20

© 2020 Daniela Rus CSAIL MIT

Challenging Interactions: Clutter, Human-Robot Systems

10/1/20

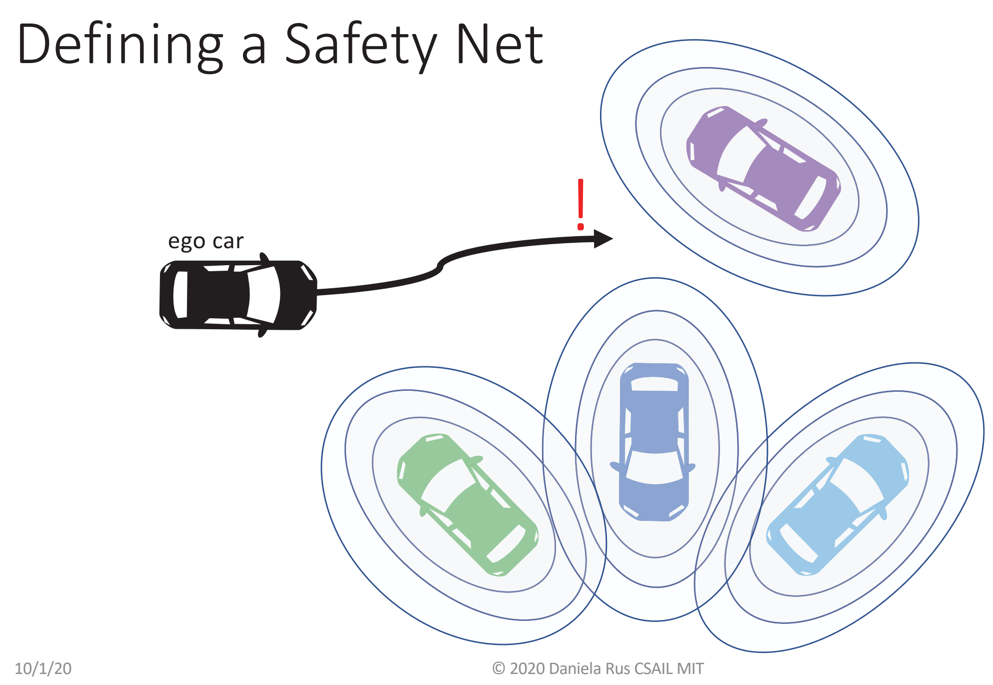
© 2020 Daniela Rus CSAIL MIT

Navigating in Clutter

[ocregister.com]

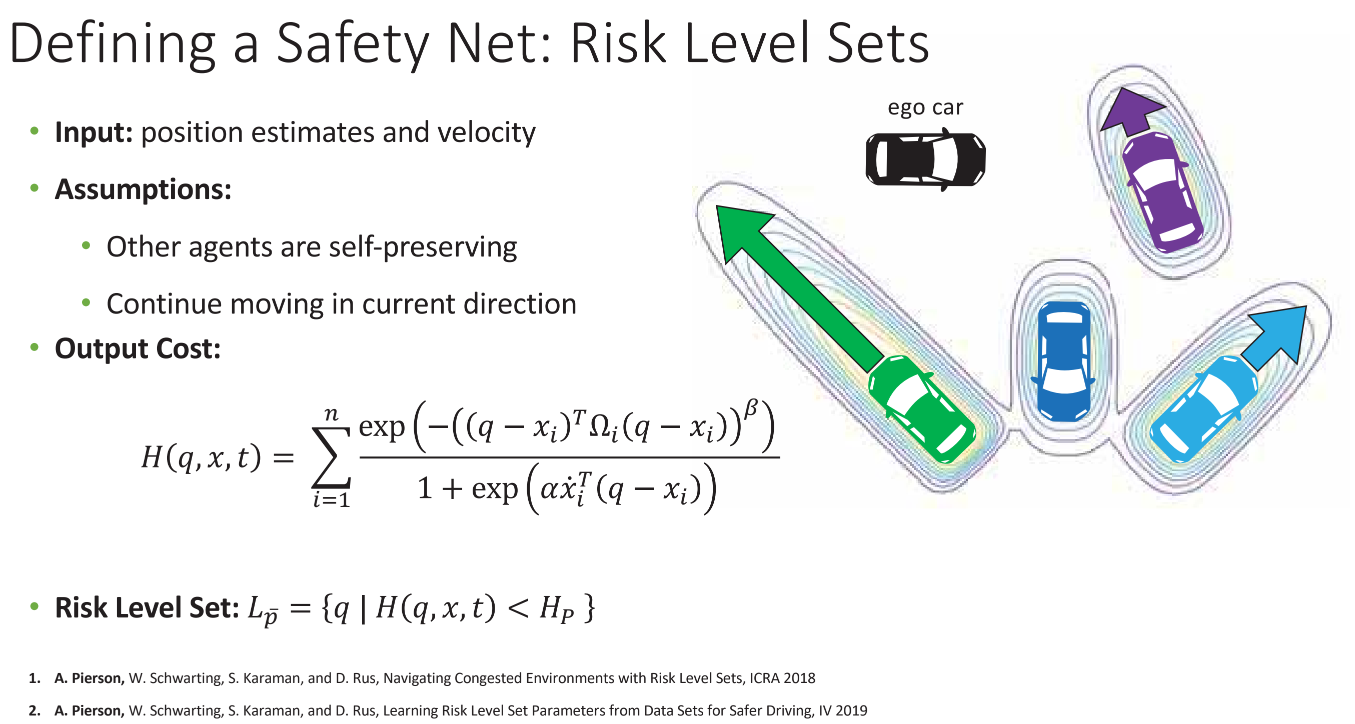
[nacto.org]

- Agnostic to static and dynamic obstacles in environment
- Use density and velocity field to compute dynamic risk density



Defining a Safety Net

10/1/20



Defining a Safety Net: Risk Level Sets

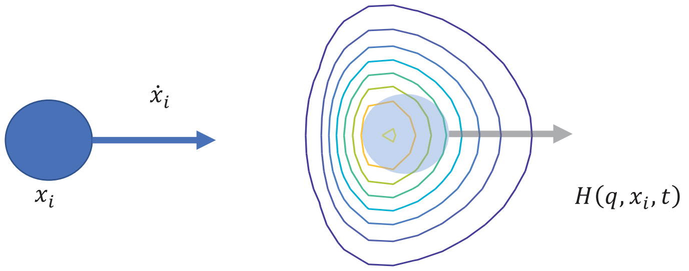
- **Input:** position estimates and velocity
- **Assumptions:**
  - Other agents are self-preserving
  - Continue moving in current direction
- **Output Cost:**

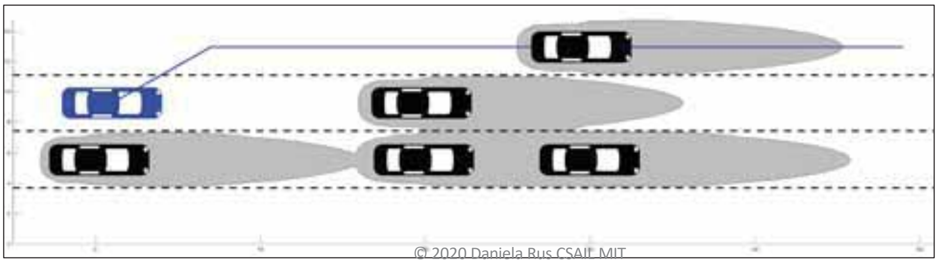
$$H(q,x,t) = \sum_{i=1}^{n} \frac{\exp\left(-\left((q-x_i)^T\Omega_i(q-x_i)\right)^{\beta}\right)}{1+\exp\left(\alpha\dot{x}_i^T(q-x_i)\right)}$$

- **Risk Level Set:** $L_{\bar{p}} = \{q \mid H(q,x,t) < H_P\}$

1. **A. Pierson,** W. Schwarting, S. Karaman, and D. Rus, Navigating Congested Environments with Risk Level Sets, ICRA 2018
2. **A. Pierson,** W. Schwarting, S. Karaman, and D. Rus, Learning Risk Level Set Parameters from Data Sets for Safer Driving, IV 2019

Planning in Congestion with Risk Level Sets

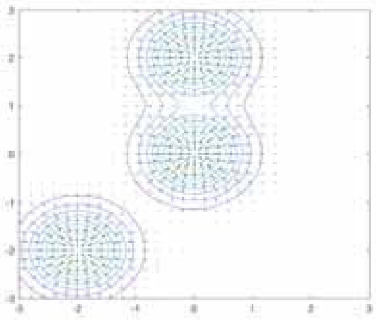




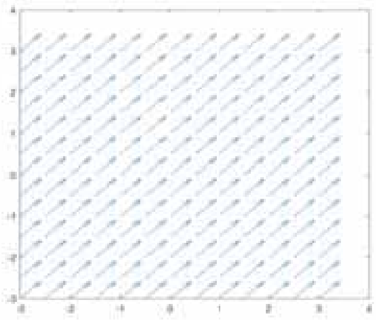
10/1/20

© 2020 Daniela Rus CSAIL MIT

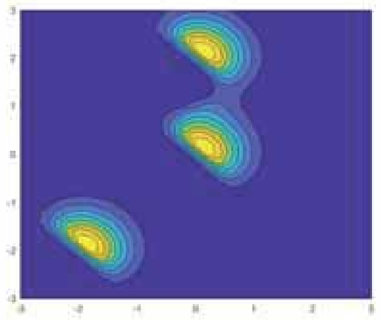
Density



Velocity Field



Dynamic Risk Density



Risk Level Sets

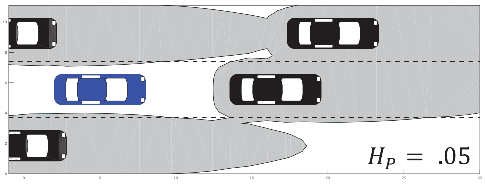
- Congestion Cost

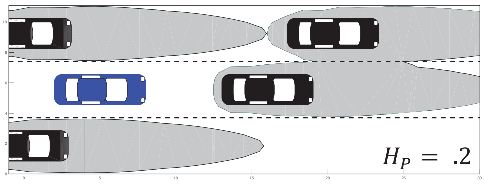
$$H(q,x,t) = \sum_{i=1}^{n} \frac{\exp\left(-\left((q-x_i)^T \Omega_i (q-x_i)\right)^{\beta}\right)}{1+\exp\left(\alpha \dot{x}_i^T (q-x_i)\right)}$$

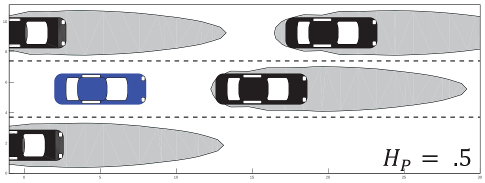
- Create level set from cost

$$L_{\bar{p}} = \{q \mid H(q,x,t) < H_P\}$$

- Plan actions within $L_{\bar{p}}$
- Higher value of H_P → higher risk (**ICRA 2018**)

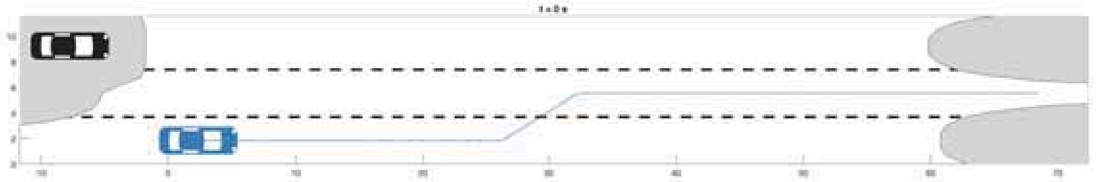






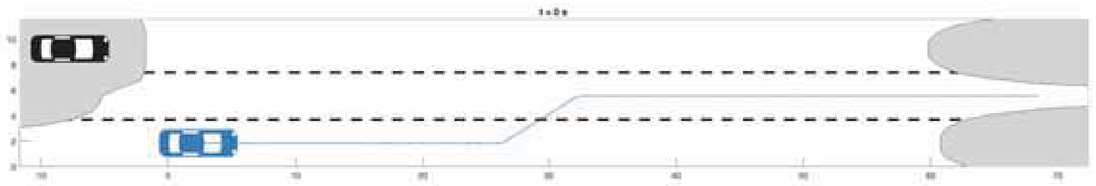
Simulation: Conservative vs Aggressive Driver

- White area: $L_{\bar{p}} = \{q \mid H(q,x,t) < H_P\}$
- Low H_P → lower risk, more conservative



Simulation: Conservative vs Aggressive Driver

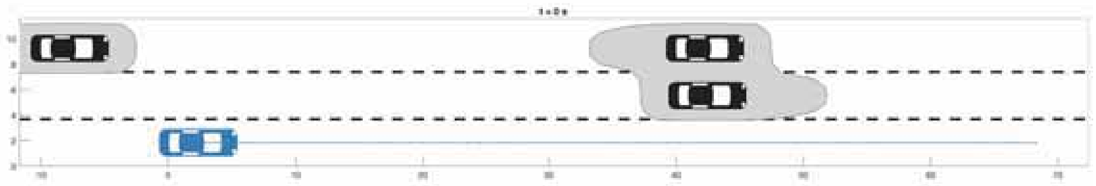
- White area: $L_{\bar{p}} = \{q \mid H(q,x,t) < H_P\}$
- Low H_P → lower risk, more conservative



10/1/20

Simulation: Conservative vs Aggressive Driver

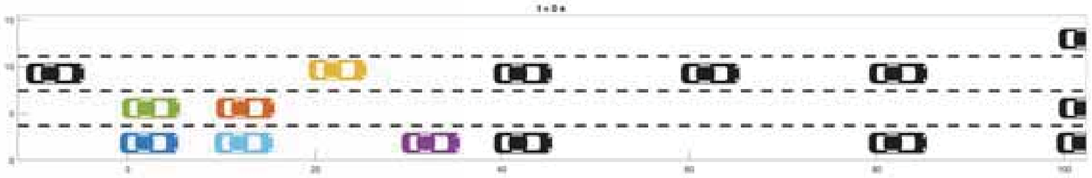
- Higher H_P → larger planning space $L_{\bar{p}}$
- More lane changes



10/1/20

Simulation: Multiple Drivers

- Each car views other cars as obstacles
- Route planner updates to other cars changing lanes



10/1/20

CARLA Validation: Risk Level Sets

- **Blue cars:** ego agents running risk level sets algorithm

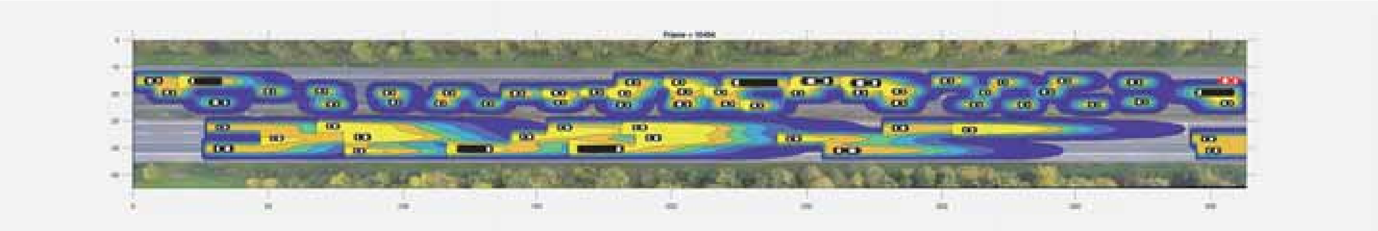
Key Features:

- Integration into our codebase across other platforms
- Multi-ego-vehicle scenarios

10/1/20

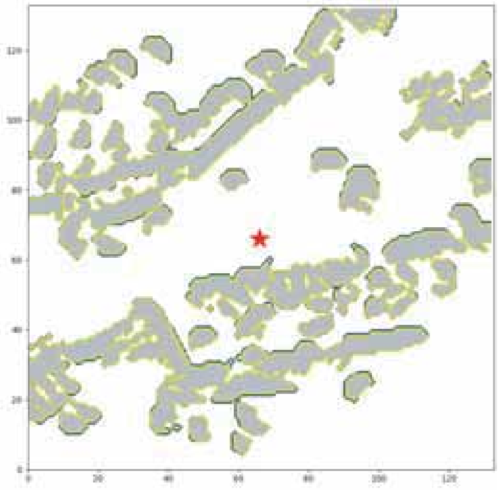
Learning Risk Level Set Parameters from Data

- NGSIM and HighD data set validation (**IV 2019**)
- Quickly identify distributions of environment and driver features



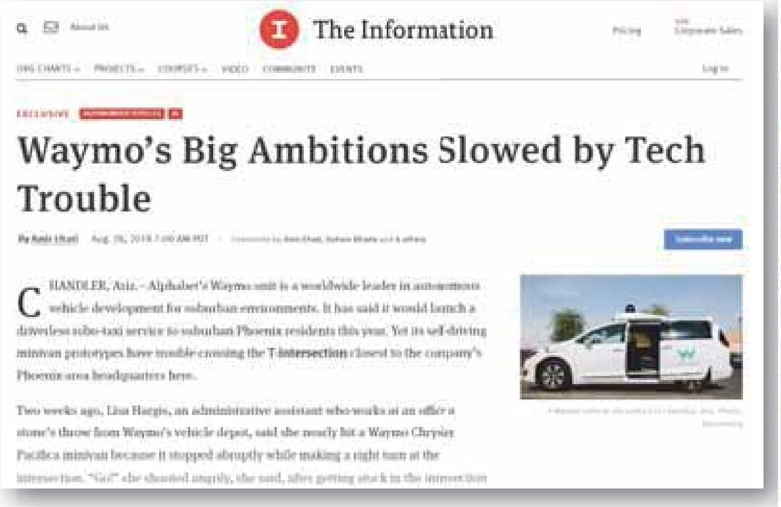
10/1/20

Risk Level Sets without Object Detection



[1] Navigating Congested Environments with Risk Level Sets, ICRA 2018, patent pending
[2] Dynamic Risk Density for Autonomous Navigation in Cluttered Environments without Object Detection, ICRA 2019, submitted

Mixed Human Driven-Robot Car Systems

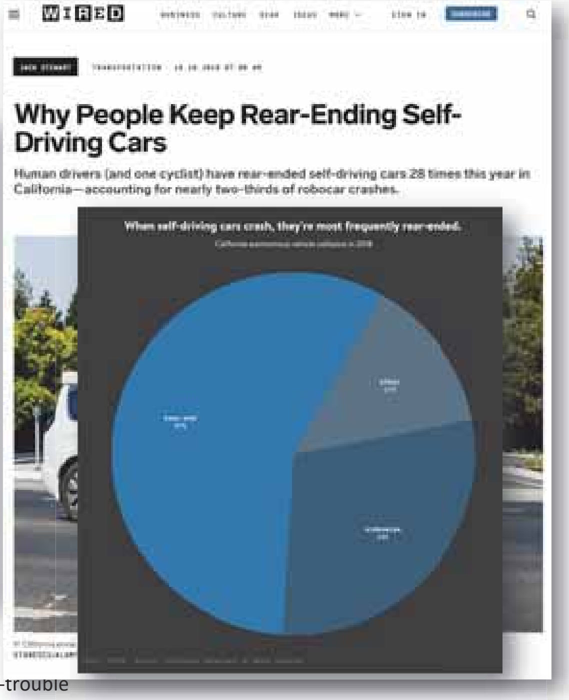


Waymo's Big Ambitions Slowed by Tech Trouble

By Amir Efrati Aug. 28, 2018 7:00 AM PDT

CHANDLER, Ariz.—Alphabet's Waymo unit is a worldwide leader in autonomous vehicle development for suburban environments. It has said it would launch a driverless robo-taxi service to suburban Phoenix residents this year. Yet its self-driving minivan prototypes have trouble crossing the T-intersection closest to the company's Phoenix area headquarters here.

Two weeks ago, Lisa Hargis, an administrative assistant who works at an office a stone's throw from Waymo's vehicle depot, said she nearly hit a Waymo Chrysler Pacifica minivan because it stopped abruptly while making a right turn at the intersection. "Go!" she shouted angrily, she said, after getting stuck in the intersection



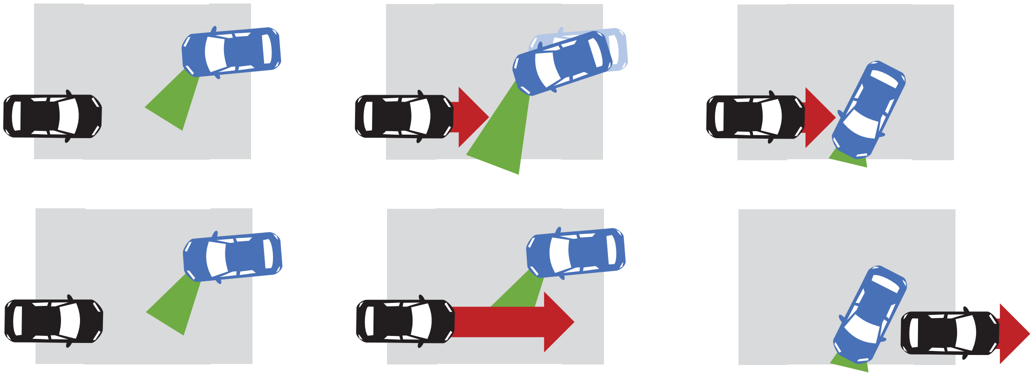
Why People Keep Rear-Ending Self-Driving Cars

Human drivers (and one cyclist) have rear-ended self-driving cars 28 times this year in California—accounting for nearly two-thirds of robocar crashes.

When self-driving cars crash, they're most frequently rear-ended.

1. https://www.theinformation.com/articles/waymos-big-ambitions-slowed-by-tech-trouble
2. https://www.wired.com/story/self-driving-car-crashes-rear-endings-why-charts-statistics/

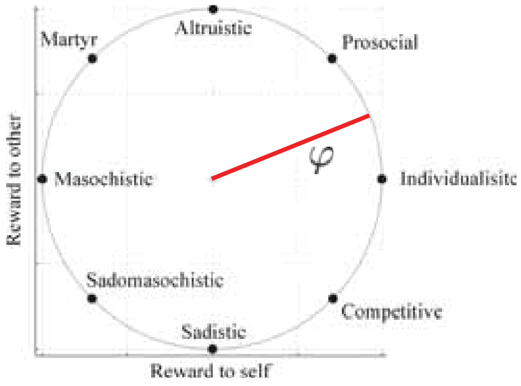
Autonomous Driving: Social Dilemma



Social dilemmas: Situations in which collective interests are at odds with private interests

Social Value Orientation (SVO) Ring

Capturing Human Preferences in Social Dilemmas



Altruistic: Maximize other party's utility, without consideration of own outcome.

Prosocial: Benefiting a group as a whole.

Individualistic: Maximize their own outcome, without concern of the utility of other agents.

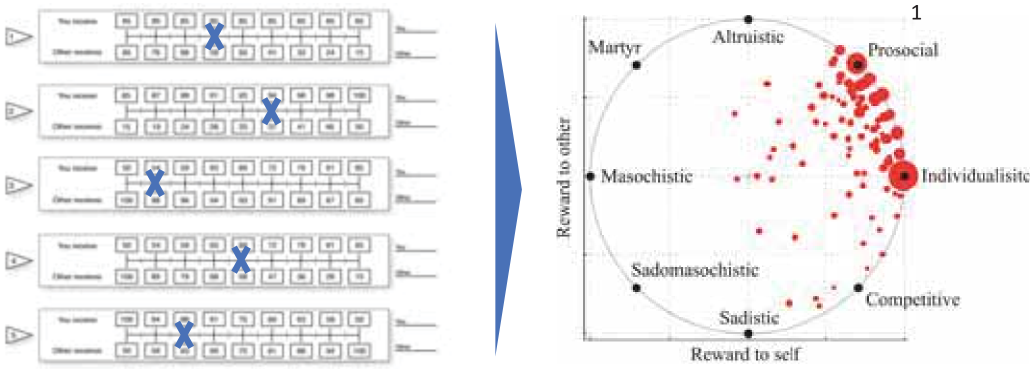
Competitive: Improve relative gain over others.

Cooperative: All agents are better off.

[1] W. B. G. Liebrand and C. G. McClintock, "The ring measure of social values: A computerized procedure for assessing individual differences in information processing and social value orientation," European Journal of Personality, vol. 2, no. 3, pp. 217–230, 1988.

Social Value Orientation (SVO) Ring

Studies of Human Preferences

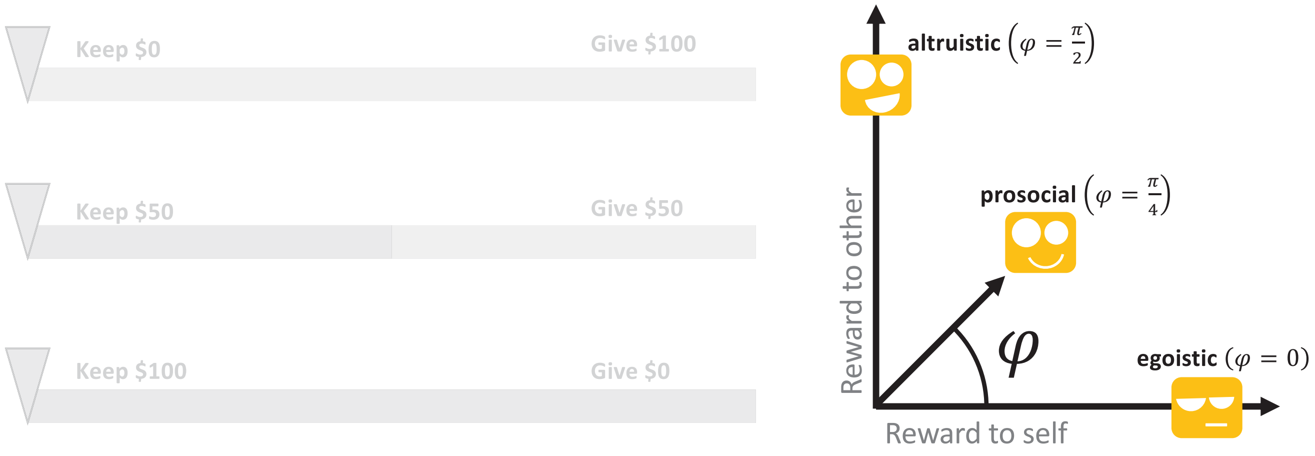


~ 90% of individuals are either prosocial (~ 50%) or individualistic (~ 40%) [2]

[1] A. Garapin, L. Muller, and B. Rahali, "Does trust mean giving and not risking? experimental evidence from the trust game," Revue d´´economie politique, vol. 125, no. 5, pp. 701–716, 2015.
[2] R. O. Murphy, K. A. Ackermann, and M. Handgraaf, "Measuring social value orientation," Judgment and Decision Making, vol. 6, no. 8, pp. 771–781, 2011.

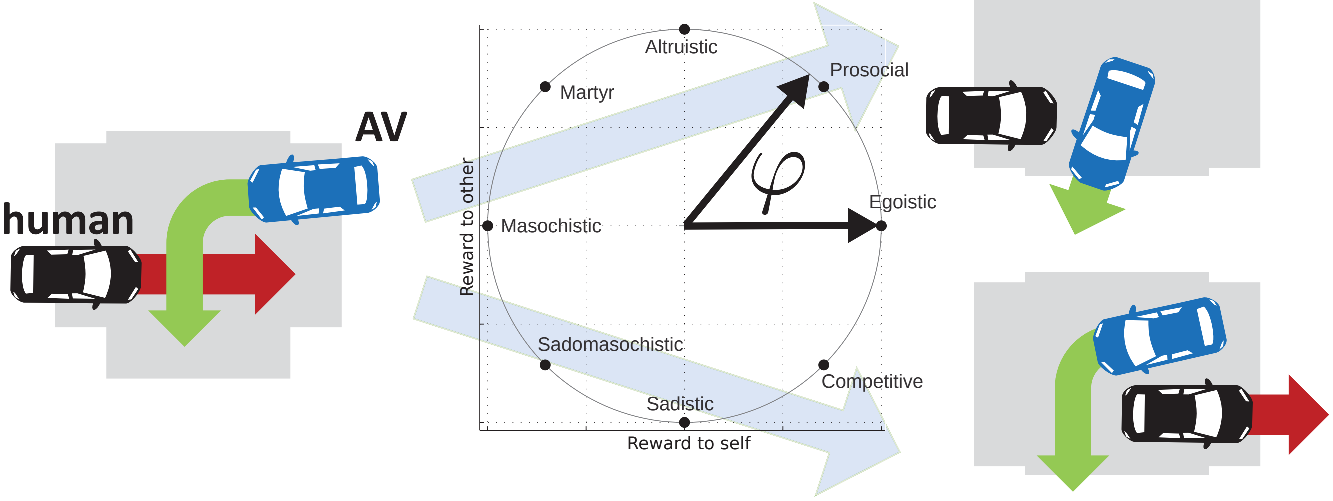
10/1/20

Split $100 with a stranger…



- A. Garapin, L. Muller, and B. Rahali, Does trust mean giving and not risking? experimental evidence from the trust game, *Revue d´´economie politique*, 2015
- R. O. Murphy, K. A. Ackermann, and M. Handgraaf, Measuring social value orientation, *Judgment and Decision Making*, 2011

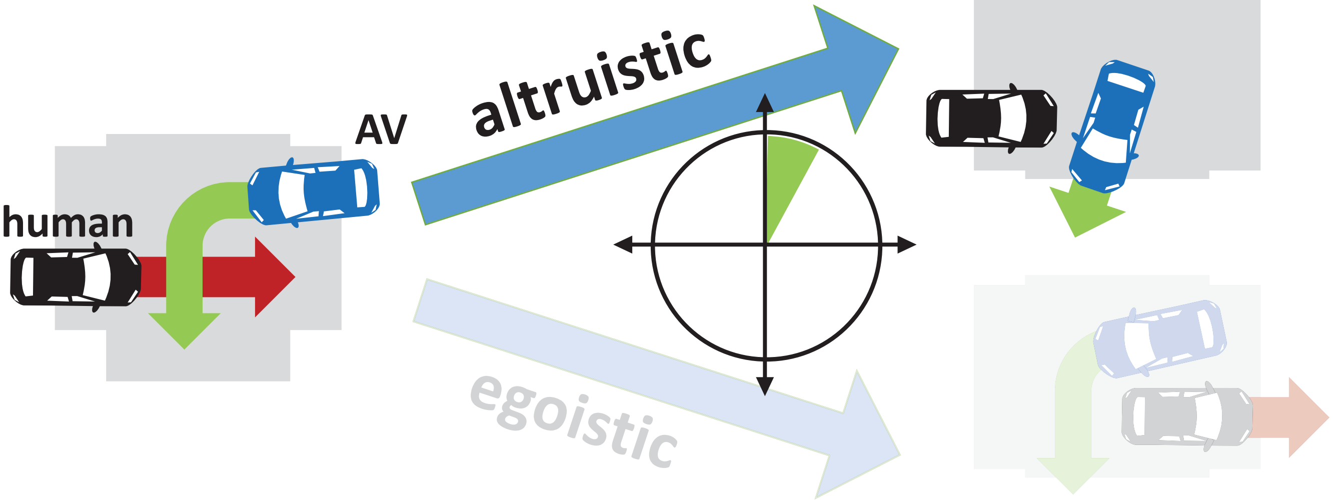
Social Value Orientation



10/1/20

© 2020 Daniela Rus CSAIL MIT

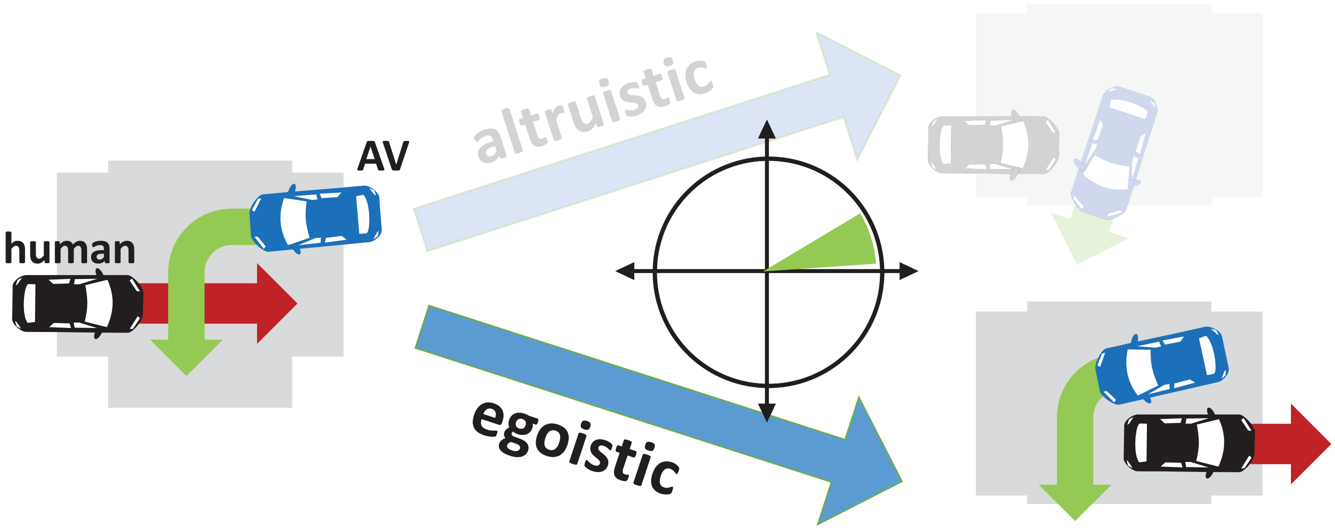
Social Value Orientation



10/1/20

© 2020 Daniela Rus CSAIL MIT

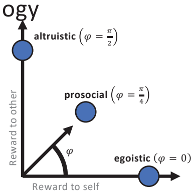
Social Value Orientation



Our Approach

Social Value Orientation

- Behavior model from social psychology


altruistic $(\varphi = \frac{\pi}{2})$
prosocial $(\varphi = \frac{\pi}{4})$
egoistic $(\varphi = 0)$
φ
Reward to other
Reward to self

$$g_i(\cdot) = \cos\varphi_i\, r_i + \sin\varphi_i\, r_j$$

- Weight reward to self vs other

Best Response Game

- Each agent maximizes its individual utility

$$G_i(\boldsymbol{x}^0,\boldsymbol{u},\varphi_i) = \sum_{k=0}^{N-1} g_i(\boldsymbol{x}^k,\boldsymbol{u}^k,\varphi_i) + g_i^N(\boldsymbol{x}^N,\varphi_i)$$

$$\boldsymbol{u}_i^* = \operatorname*{argmax}_{\boldsymbol{u}_i} G_i(\boldsymbol{x}^0,\boldsymbol{u}_i,\boldsymbol{u}_{\neg i},\varphi_i)$$

- Solve for Nash Equilibrium

Learned Rewards

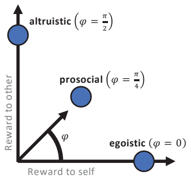
- Inverse Reinforcement Learning
- Calibrate rewards on NGSIM data set

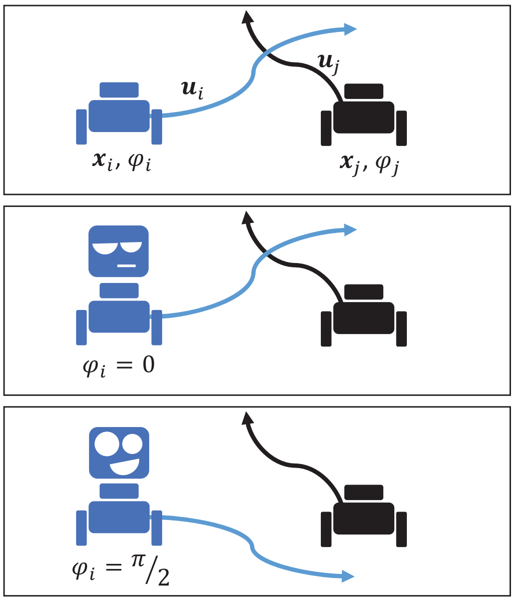
[1] W. Schwarting, **A. Pierson**, J. Alonso-Mora, S. Karaman, and D. Rus, Social Behavior for Autonomous Vehicles. *Proceedings of the National Academy of Sciences (PNAS), 2019*

Utility-Maximizing Policy with SVO

- Joint reward weighted by SVO

$$g_i(\cdot) = \underline{\cos\varphi_i\, r_i} + \underline{\sin\varphi_i\, r_j}$$


altruistic $\left(\varphi = \frac{\pi}{2}\right)$
prosocial $\left(\varphi = \frac{\pi}{4}\right)$
egoistic $(\varphi = 0)$
φ
Reward to other
Reward to self


$\boldsymbol{u}_i$
$\boldsymbol{u}_j$
$\boldsymbol{x}_i, \varphi_i$
$\boldsymbol{x}_j, \varphi_j$
$\varphi_i = 0$
$\varphi_i = {}^{\pi}/_{2}$

- Utility over time horizon

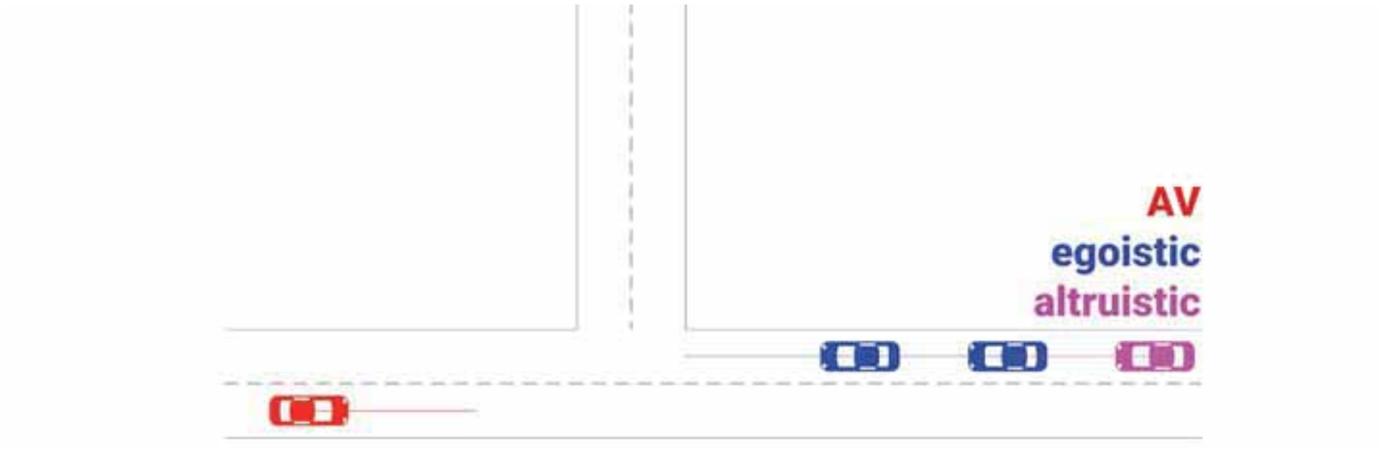
$$G_i(\boldsymbol{x}^0, \boldsymbol{u}, \varphi_i) = \sum_{k=0}^{N-1} g_i(\boldsymbol{x}^k, \boldsymbol{u}^k, \varphi_i) + g_i^N(\boldsymbol{x}^N, \varphi_i)$$

- Find control $\boldsymbol{u}_i^*$ that maximizes utility

$$\boldsymbol{u}_i^* = \underset{\boldsymbol{u}_i}{\operatorname{argmax}}\, G_i(\boldsymbol{x}^0, \boldsymbol{u}_i, \boldsymbol{u}_{\neg i}, \varphi_i)$$

Unprotected Left Turns

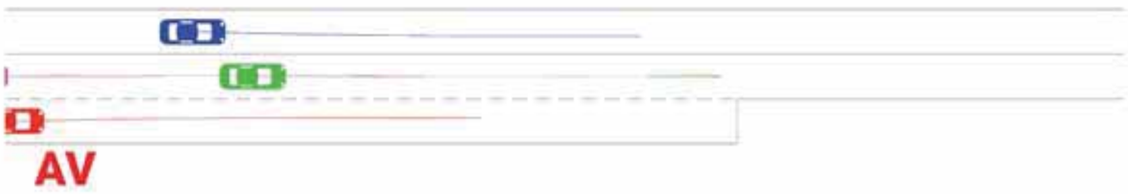
The AV must wait for an altruistic driver to yield



1. W. Schwarting, **A. Pierson**, J. Alonso-Mora, S. Karaman, and D. Rus, Social Behavior for Autonomous Vehicles. *Proceedings of the National Academy of Sciences (PNAS), 2019*

Egoistic Merge

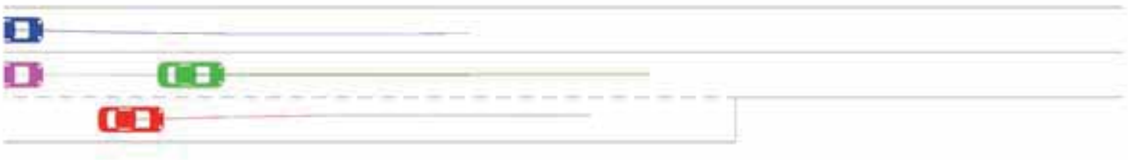
Among egoistic drivers, the AV must wait to merge



1. W. Schwarting, **A. Pierson**, J. Alonso-Mora, S. Karaman, and D. Rus, Social Behavior for Autonomous Vehicles. *Proceedings of the National Academy of Sciences (PNAS), 2019*

Prosocial Merge

Prosocial drivers create a gap for the AV to merge

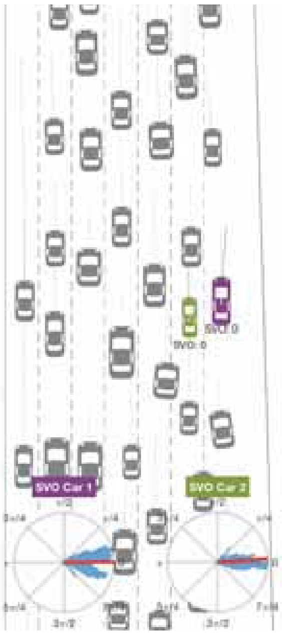


1. W. Schwarting, **A. Pierson**, J. Alonso-Mora, S. Karaman, and D. Rus, Social Behavior for Autonomous Vehicles. *Proceedings of the National Academy of Sciences (PNAS), 2019*

Estimate SVO online

Estimate SVO of other drivers online

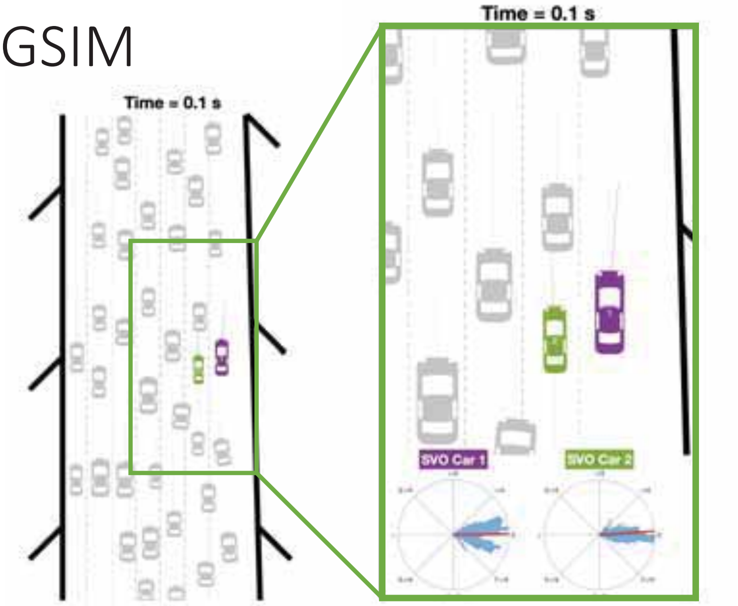
Integrate into motion planner to improve decision-making and predictions



10/1/20

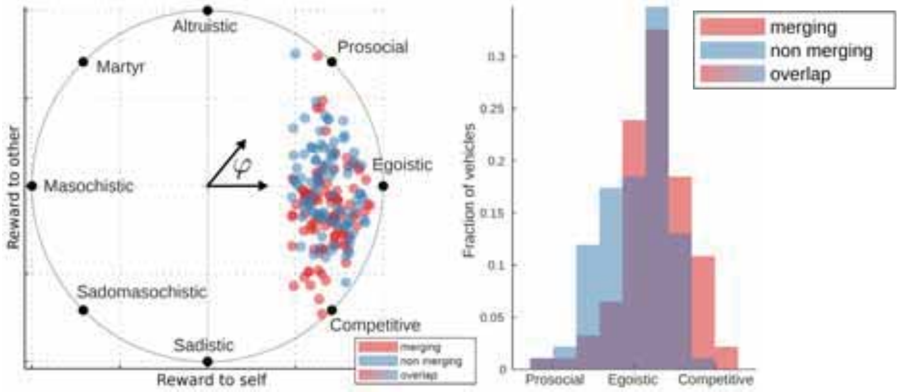
SVO Predictions on NGSIM

Estimating driver SVO improves trajectory predictions by 25%



1. W. Schwarting, **A. Pierson**, J. Alonso-Mora, S. Karaman, and D. Rus, Social Behavior for Autonomous Vehicles. *Proceedings of the National Academy of Sciences (PNAS), 2019*

SVO Trends in NGSIM dataset

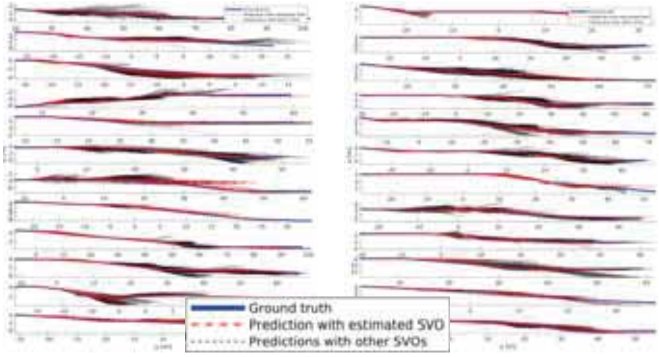


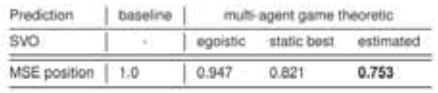
Merging vehicles are more competitive than non merging vehicles ($p < 0.002$)

1. W. Schwarting, **A. Pierson**, J. Alonso-Mora, S. Karaman, and D. Rus, Social Behavior for Autonomous Vehicles. *Proceedings of the National Academy of Sciences (PNAS), 2019*

Evaluation of SVO on NGSIM dataset

Improved prediction with dynamically estimated SVO during merges





Prediction	baseline	multi-agent game theoretic		
SVO	-	egoistic	static best	estimated
MSE position	1.0	0.947	0.821	**0.753**

Table 1. Relative mean square position error (MSE) between predicted and actual trajectories, as compared to a single-agent planning baseline.

25% reduced prediction error

10/1/20

© 2020 Daniela Rus CSAIL MIT

10/1/20

© 2020 Daniela Rus CSAIL MIT

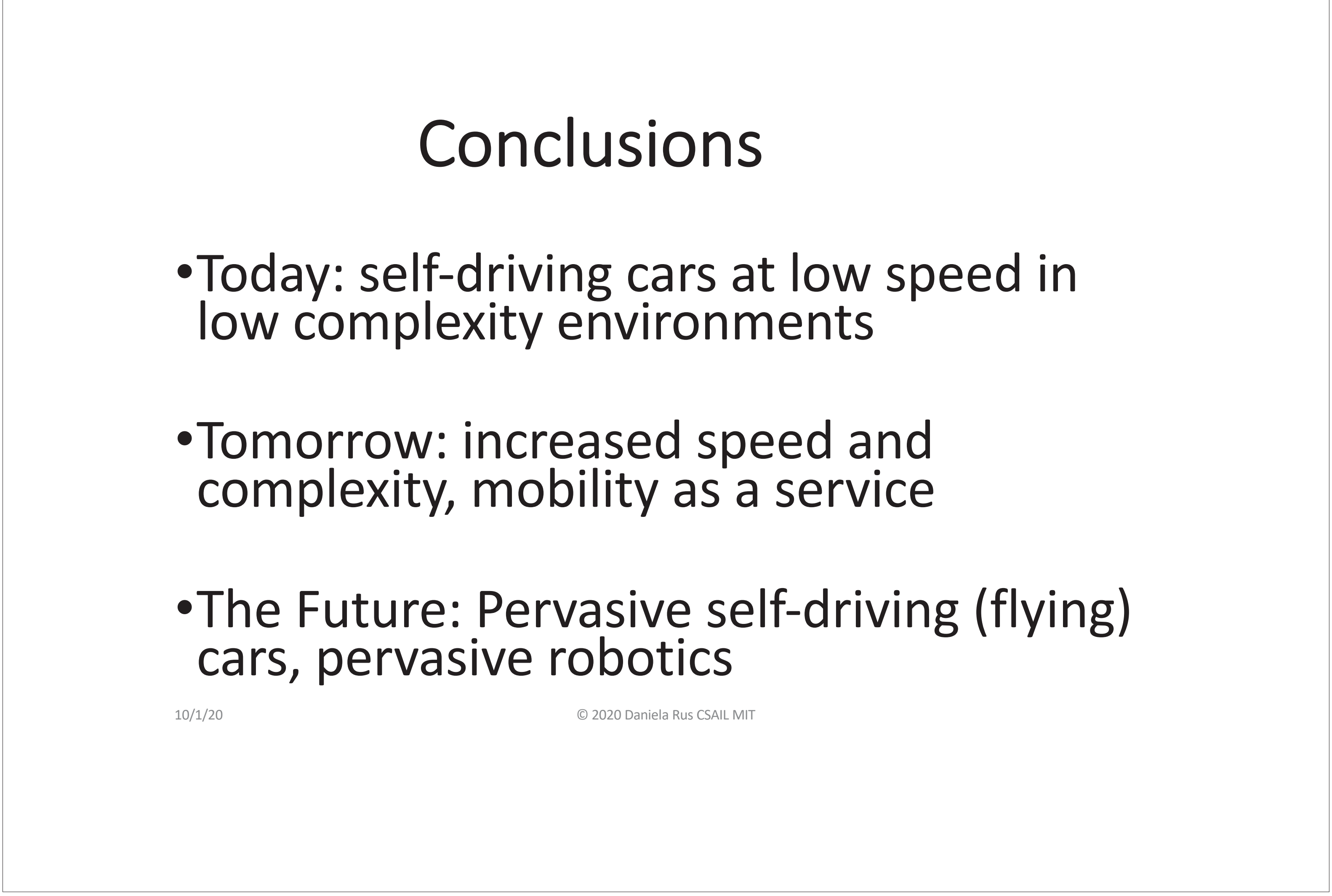
10/1/20 © 2020 Daniela Rus CSAIL MIT

What will come first?

Level 5 Autonomy, or

The Flying car?

10/1/20 © 2020 Daniela Rus CSAIL MIT



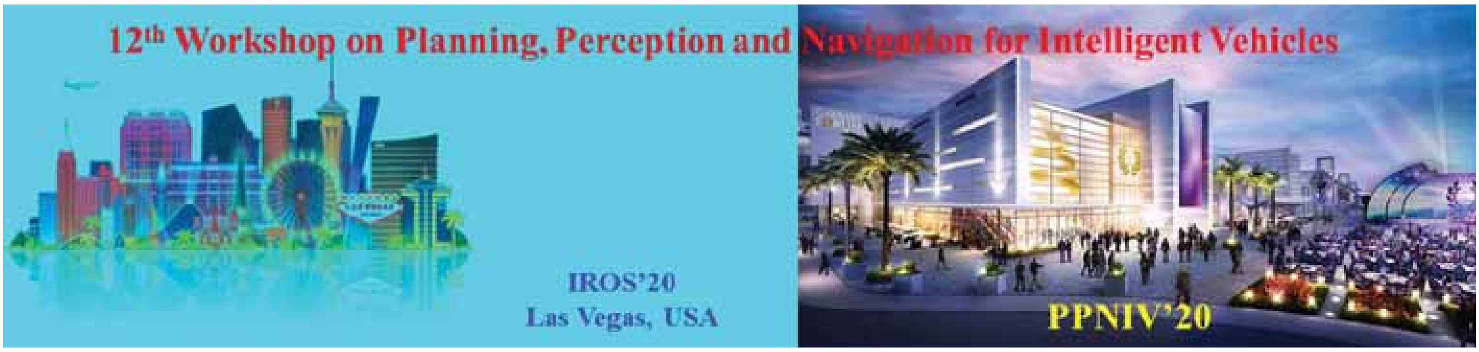
Conclusions

- Today: self-driving cars at low speed in low complexity environments

- Tomorrow: increased speed and complexity, mobility as a service

- The Future: Pervasive self-driving (flying) cars, pervasive robotics

10/1/20


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

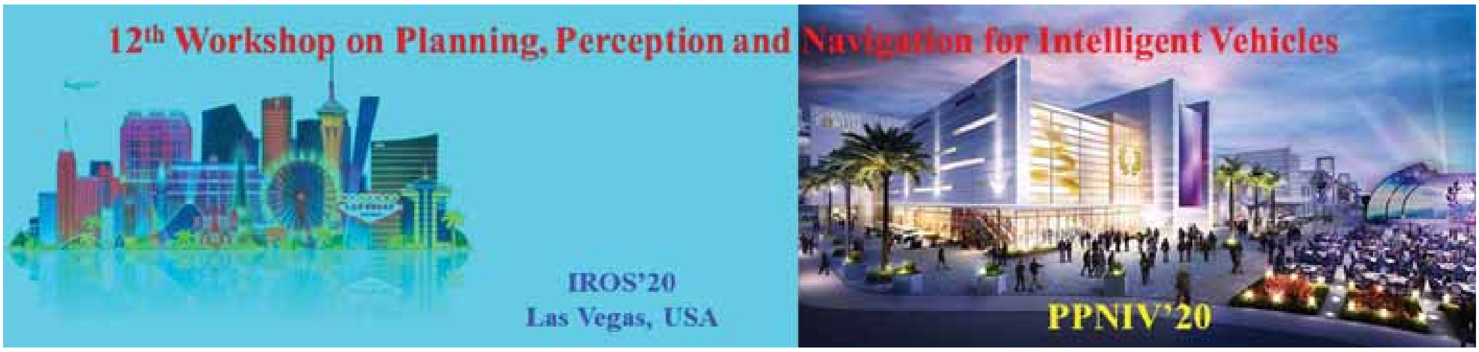
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Keynote speaker: **Mohan M Trivedi**
(University of California, USA)

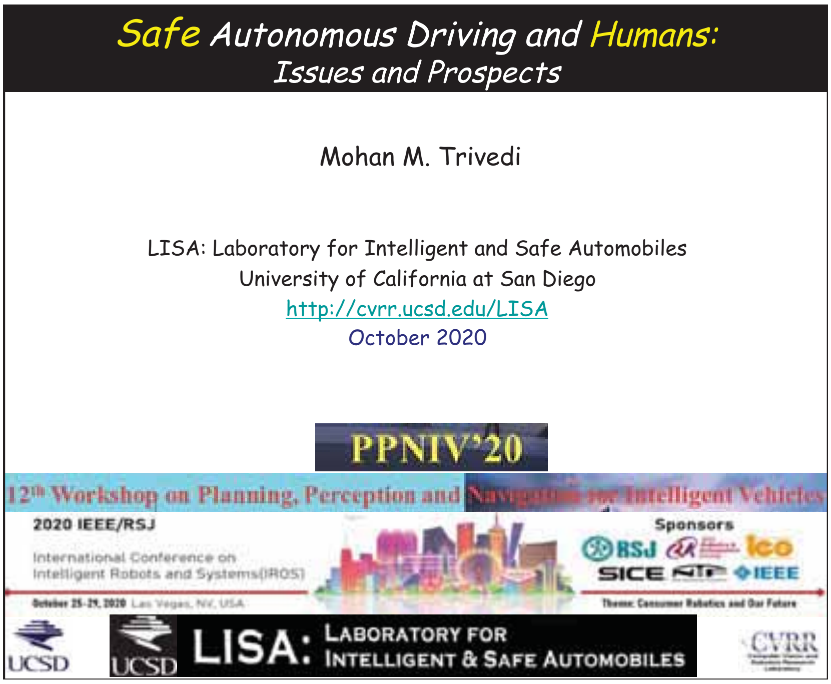
Safe Autonomous Driving and Humans: Perception and Transitions

Abstract: These are truly exciting times especially for researchers and scholars active in robotics and intelligent systems fields. Fruits of their labor are enabling transformative changes in daily lives of general public. In this presentation we will focus on changes affecting our mobility on roads with highly automated intelligent vehicles. We specifically discuss issues related to the understanding of human agents interacting with the automated vehicle, either as occupants of such vehicles, or who are in the near vicinity of the vehicles, as pedestrians, cyclists, or inside surrounding vehicles. These issues require deeper examination and careful resolution to assure safety, reliability and robustness of these highly complex systems for operation on public roads. The presentation will highlight recent research dealing with understanding of activities, behavior, intentions of humans specifically in the context of autonomous driving and transition controls.

Biograpghy: Mohan Trivedi is a Distinguished Professor of Engineering and founding director of the Computer Vision and Robotics Research Laboratory, as well as the Laboratory for Intelligent and Safe Automobiles (LISA) at the University of California San Diego. These labs have played significant roles in the development of human-centered safe autonomous driving, advanced driver assistance systems, vision systems for intelligent transportation, homeland security, assistive technologies and human-robot interaction fields. Trivedi has received the IEEE Intelligent Transportation Systems (ITS) Society's Outstanding Researcher Award and LEAD Institution Award, as well as the Meritorious Service Award of the IEEE Computer Society. He is a Fellow of IEEE, SPIE, and IAPR. He serves very regularly as a consultant to industry and government agencies in the USA and abroad. Trivedi frequently participates on panels dealing with technological, strategic, privacy, and ethical issues surrounding research areas he is involved in.


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems



Safe Autonomous Driving and Humans:
Issues and Prospects

Mohan M. Trivedi

LISA: Laboratory for Intelligent and Safe Automobiles
University of California at San Diego
http://cvrr.ucsd.edu/LISA
October 2020

PPNIV'20

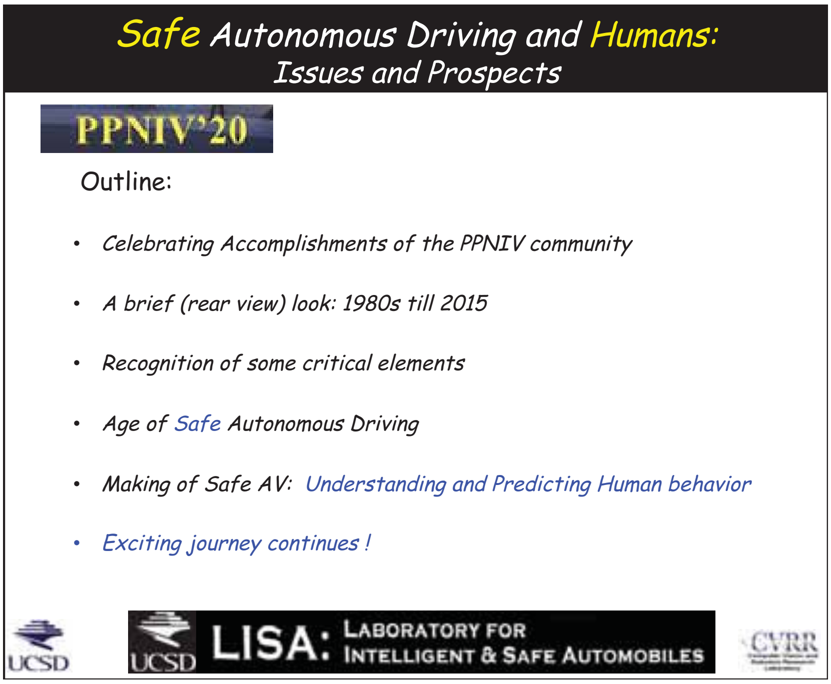
12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2020 IEEE/RSJ
International Conference on
Intelligent Robots and Systems(IROS)
October 25-29, 2020 Las Vegas, NV, USA

Sponsors

Theme: Consumer Robotics and Our Future

RSJ
SICE
IEEE
UCSD
LISA: Laboratory for Intelligent & Safe Automobiles
CVRR



Safe Autonomous Driving and Humans:
Issues and Prospects

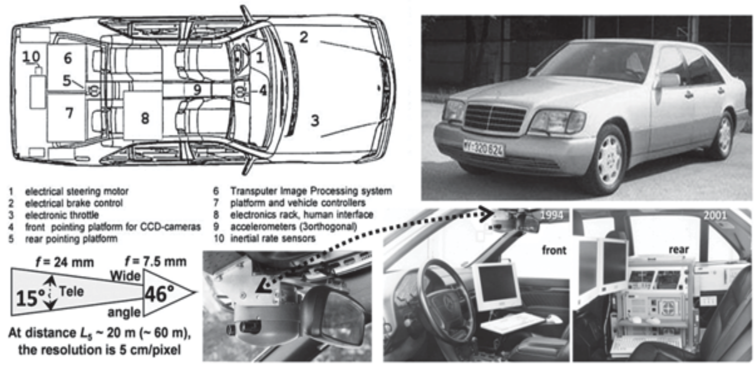
PPNIV'20

Outline:

- *Celebrating Accomplishments of the PPNIV community*
- *A brief (rear view) look: 1980s till 2015*
- *Recognition of some critical elements*
- *Age of Safe Autonomous Driving*
- *Making of Safe AV: Understanding and Predicting Human behavior*
- *Exciting journey continues !*

UCSD
LISA: Laboratory for Intelligent & Safe Automobiles
CVRR

Vision for Intelligent Vehicles: Past 1980-2000


10
6
5
7
8
9
1
2
4
3
1 electrical steering motor
2 electrical brake control
3 electronic throttle
4 front pointing platform for CCD-cameras
5 rear pointing platform
6 Transputer Image Processing system
7 platform and vehicle controllers
8 electronics rack, human interface
9 accelerometers (3orthogonal)
10 inertial rate sensors
f = 24 mm
f = 7.5 mm
Wide
15°
Tele
46°
angle
At distance L_5 ~ 20 m (~ 60 m), the resolution is 5 cm/pixel
1994
front
2001
rear

Ernst Dickmanns, "The development of machine vision for road vehicles in the last decade." *IEEE Intelligent Vehicles Symposium,* 2002

Vision for Intelligent Vehicles: Past 1980-2000

Days 6-7-8: July 28th, 29th, 30th, 1995 Las Vegas and drive to San Diego

Driving Time: 9:00am - 1:45pm

```
Segment Autonomy Stats:
    Autonomous Driving Percentage:  99.8%  (322.99 / 323.5 miles)
    Longest Autonomous Segment:     59.36 miles
    Avg Speed:                      70.37 mph

Total Autonomous Stats (including Pittsburgh to Washington D.C. segment):
    Autonomous Driving Percentage:  98.2%  (2796.87 / 2849.13 miles)
    Longest Autonomous Segment:     68.98 miles
    Avg Speed:                      63.8 mph
```

Todd Jochem, Dean Pomerleau, Charles Thorpe, "Vision Guided Lane Transition," *IEEE Intelligent Vehicles,* 1995.

Trivedi_IROS-PPNIV20

Seeking Full Autonomy, *2003-2009*

DARPA Grand Challenge I
Barstow to Primm
March 13, 2004

142 miles
10 hours

DARPA Grand Challenge II
Desert Classic
October 8, 2005

132 miles
10 hours

DARPA Grand Challenge III
Urban Challenge
November 3, 2007

60 mi
6 hou

2014 Birth of a new age ?

Vernetzte Sensorik des Forschungsfahrzeugs S 500 INTELLIGENT DRIVE
Hochautomatisiertes Fahren auf Basis von zusätzlichen seriennahen Sensoren

Self Driving Cars 2015

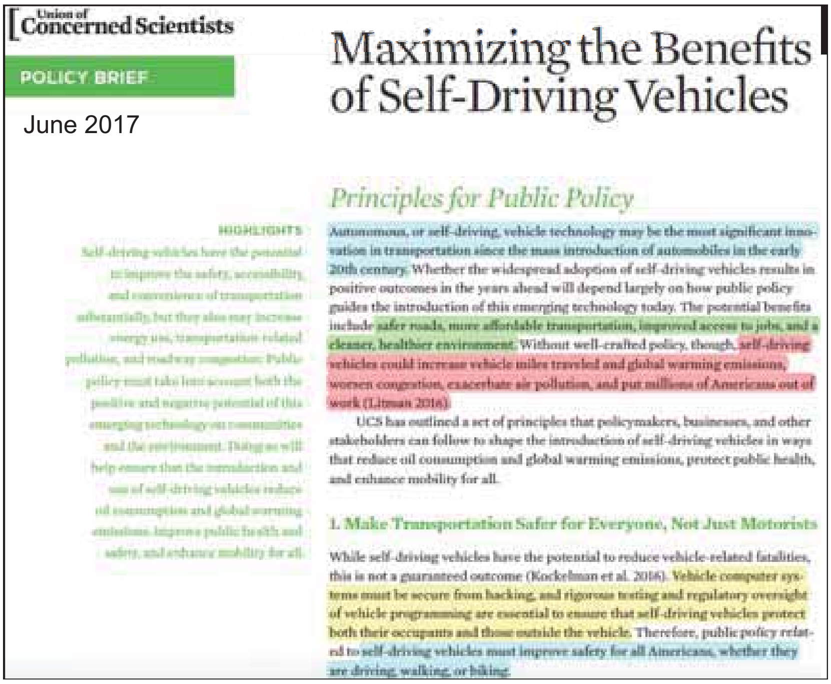
CBS THIS MORNING
NO HANDS
60 MINUTES TEST RIDES A SELF-DRIVING CAR
60 MINUTES
UCSD
UCSD
LISA: LABORATORY FOR INTELLIGENT & SAFE AUTOMOBILES
CVRR

Quest for Fully Autonomous Driving

Seen this?

Trivedi_IROS-PPNIV20

4



Union of Concerned Scientists

POLICY BRIEF

June 2017

Maximizing the Benefits of Self-Driving Vehicles

Principles for Public Policy

HIGHLIGHTS

Self-driving vehicles have the potential to improve the safety, accessibility, and convenience of transportation substantially, but they also may increase energy use, transportation-related pollution, and roadway congestion. Public policy must take into account both the positive and negative potential of this emerging technology on communities and the environment. Doing so will help ensure that the introduction and use of self-driving vehicles reduce oil consumption and global warming emissions, improve public health and safety, and enhance mobility for all.

Autonomous, or self-driving, vehicle technology may be the most significant innovation in transportation since the mass introduction of automobiles in the early 20th century. Whether the widespread adoption of self-driving vehicles results in positive outcomes in the years ahead will depend largely on how public policy guides the introduction of this emerging technology today. The potential benefits include safer roads, more affordable transportation, improved access to jobs, and a cleaner, healthier environment. Without well-crafted policy, though, self-driving vehicles could increase vehicle miles traveled and global warming emissions, worsen congestion, exacerbate air pollution, and put millions of Americans out of work (Litman 2016).

UCS has outlined a set of principles that policymakers, businesses, and other stakeholders can follow to shape the introduction of self-driving vehicles in ways that reduce oil consumption and global warming emissions, protect public health, and enhance mobility for all.

1. Make Transportation Safer for Everyone, Not Just Motorists

While self-driving vehicles have the potential to reduce vehicle-related fatalities, this is not a guaranteed outcome (Kockelman et al. 2016). Vehicle computer systems must be secure from hacking, and rigorous testing and regulatory oversight of vehicle programming are essential to ensure that self-driving vehicles protect both their occupants and those outside the vehicle. Therefore, public policy related to self-driving vehicles must improve safety for all Americans, whether they are driving, walking, or biking.

Towards Human-Centered Autonomous Driving

What happens if the vehicle makes a mistake?

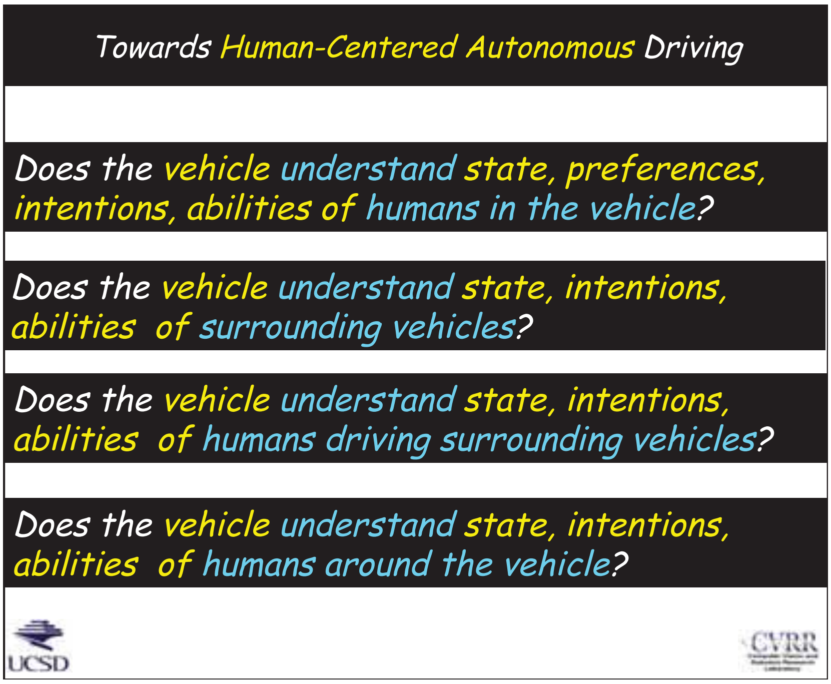
What happens if the vehicle doesn't know it made a mistake?

What happens if the vehicle refuses to let go ?

UCSD

CVRR

Trivedi_IROS-PPNIV20



Towards Human-Centered Autonomous Driving

Does the vehicle understand state, preferences, intentions, abilities of humans in the vehicle?

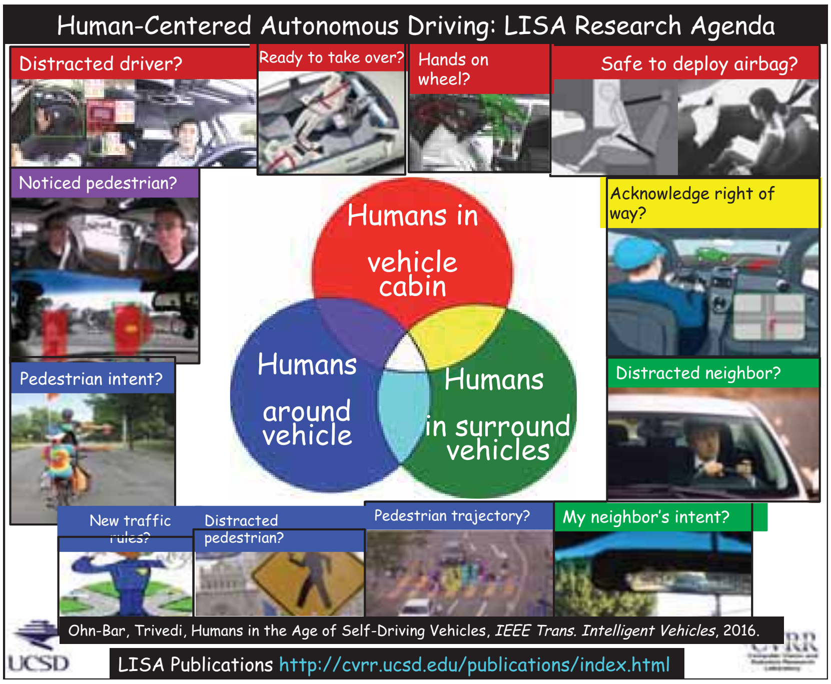
Does the vehicle understand state, intentions, abilities of surrounding vehicles?

Does the vehicle understand state, intentions, abilities of humans driving surrounding vehicles?

Does the vehicle understand state, intentions, abilities of humans around the vehicle?

UCSD

CVRR



Human-Centered Autonomous Driving: LISA Research Agenda

Distracted driver?

Ready to take over?

Hands on wheel?

Safe to deploy airbag?

Noticed pedestrian?

Acknowledge right of way?

Humans in vehicle cabin

Humans around vehicle

Humans in surround vehicles

Pedestrian intent?

Distracted neighbor?

New traffic rules?

Distracted pedestrian?

Pedestrian trajectory?

My neighbor's intent?

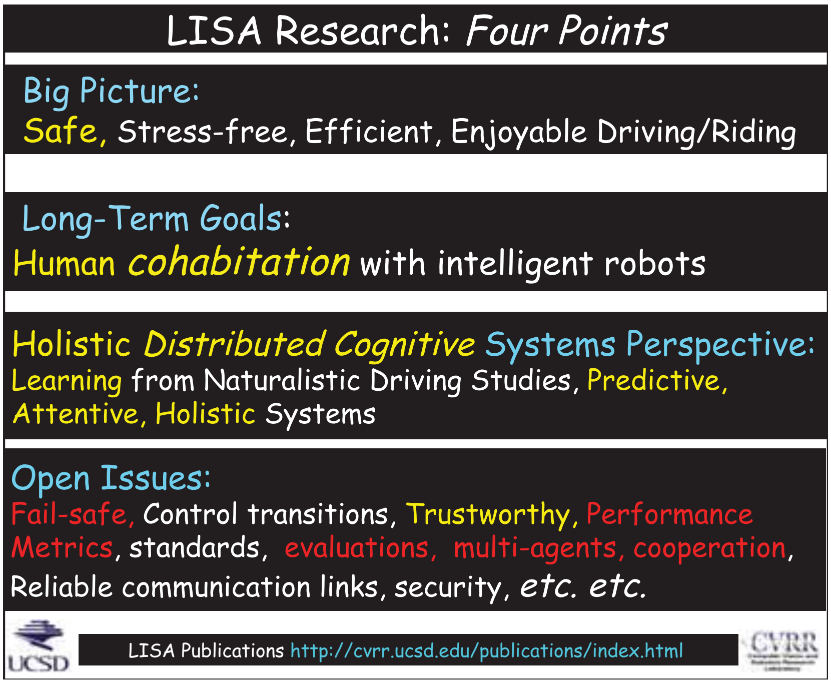
Ohn-Bar, Trivedi, Humans in the Age of Self-Driving Vehicles, *IEEE Trans. Intelligent Vehicles*, 2016.

LISA Publications http://cvrr.ucsd.edu/publications/index.html

UCSD

CVRR

Trivedi_IROS-PPNIV20



LISA Research: *Four Points*

Big Picture:
Safe, Stress-free, Efficient, Enjoyable Driving/Riding

Long-Term Goals:
Human *cohabitation* with intelligent robots

Holistic *Distributed Cognitive* Systems Perspective:
Learning from Naturalistic Driving Studies, Predictive, Attentive, Holistic Systems

Open Issues:
Fail-safe, Control transitions, Trustworthy, Performance Metrics, standards, evaluations, multi-agents, cooperation, Reliable communication links, security, *etc. etc.*

UCSD

LISA Publications http://cvrr.ucsd.edu/publications/index.html

CVRR

LISA-T: for Safe Autonomous Driving

LISA-T: for Safe Autonomous Driving

Key Research Contributors

Akshay Rangesh

Kevan Yuen

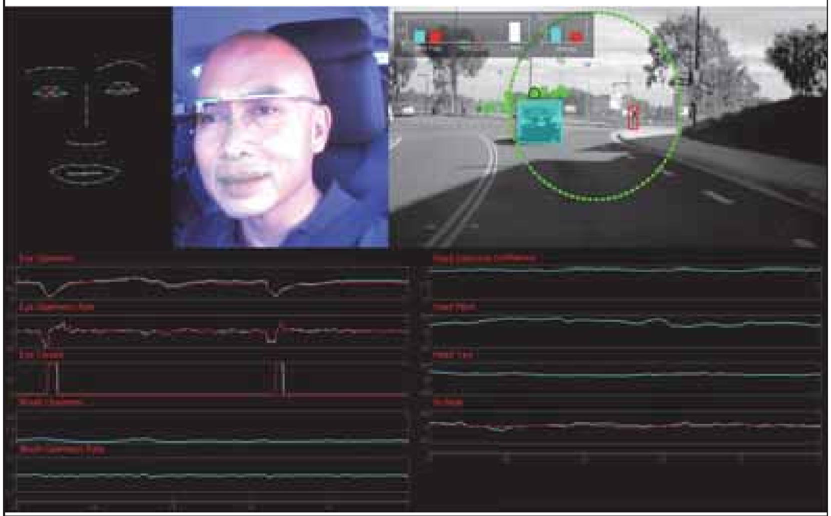
Nachiket Deo

LISA-T: for Safe Autonomous Driving

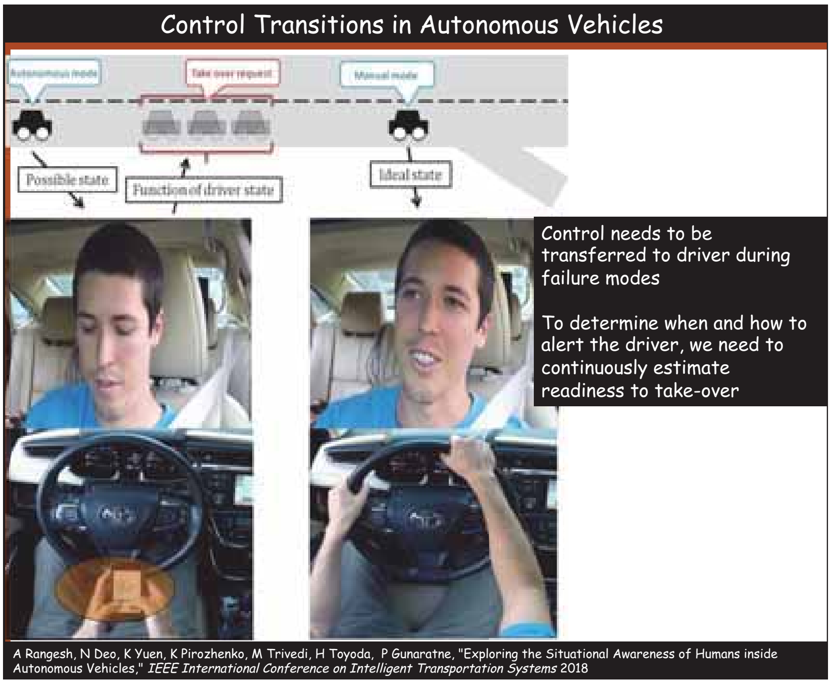
Continuous Situational Awareness

A Rangesh, N Deo, K Yuen, K Pirozhenko, M Trivedi, H Toyoda, P Gunaratne, "Exploring the Situational Awareness of Humans inside Autonomous Vehicles," *IEEE International Conference on Intelligent Transportation Systems* 2018

Continuous Situational Awareness



Ashish Tawari, Andreas Mogelmose, Sujitha Martin, Thomas Moeslund, and Mohan M. Trivedi, "Attention Estimation by Simultaneous Analysis of Viewer and View," IEEE Intelligent Transportation Systems

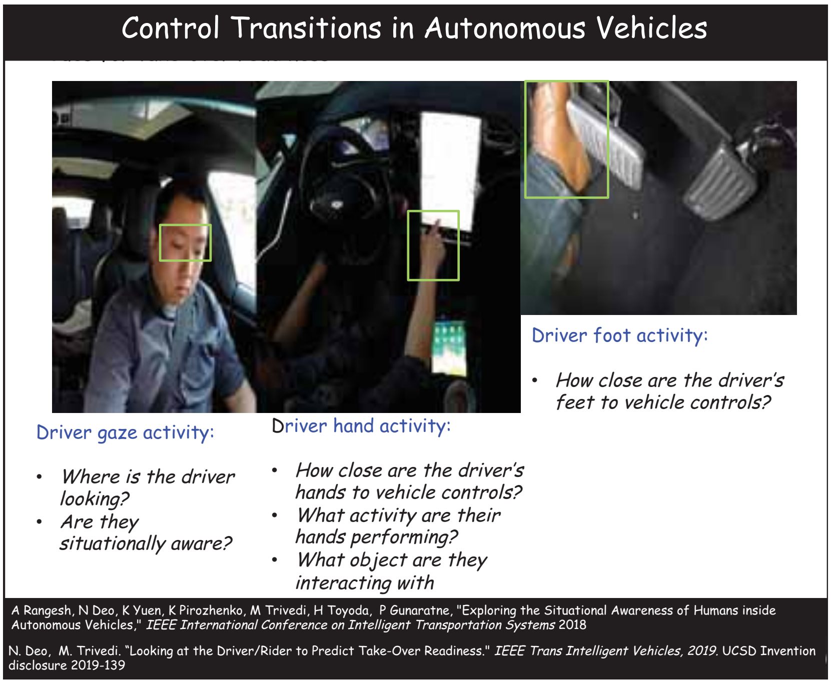

Autonomous mode
Take over request
Manual mode
Possible state
Function of driver state
Ideal state

Control Transitions in Autonomous Vehicles

Control needs to be transferred to driver during failure modes

To determine when and how to alert the driver, we need to continuously estimate readiness to take-over

A Rangesh, N Deo, K Yuen, K Pirozhenko, M Trivedi, H Toyoda, P Gunaratne, "Exploring the Situational Awareness of Humans inside Autonomous Vehicles," *IEEE International Conference on Intelligent Transportation Systems* 2018



Control Transitions in Autonomous Vehicles

Driver gaze activity:

- *Where is the driver looking?*
- *Are they situationally aware?*

Driver hand activity:

- *How close are the driver's hands to vehicle controls?*
- *What activity are their hands performing?*
- *What object are they interacting with*

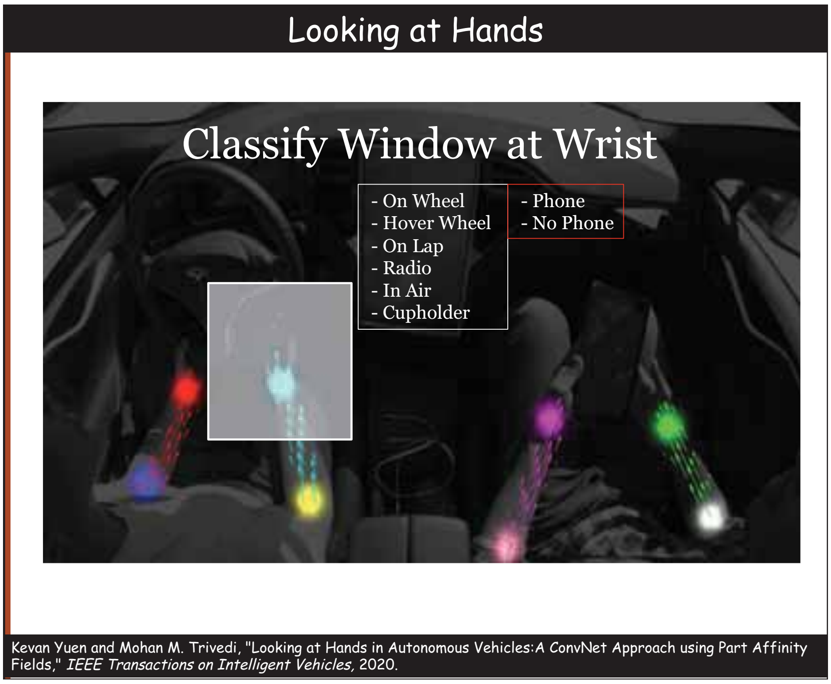
Driver foot activity:

- *How close are the driver's feet to vehicle controls?*

A Rangesh, N Deo, K Yuen, K Pirozhenko, M Trivedi, H Toyoda, P Gunaratne, "Exploring the Situational Awareness of Humans inside Autonomous Vehicles," *IEEE International Conference on Intelligent Transportation Systems* 2018

N. Deo, M. Trivedi. "Looking at the Driver/Rider to Predict Take-Over Readiness." *IEEE Trans Intelligent Vehicles, 2019.* UCSD Invention disclosure 2019-139

Trivedi_IROS-PPNIV20



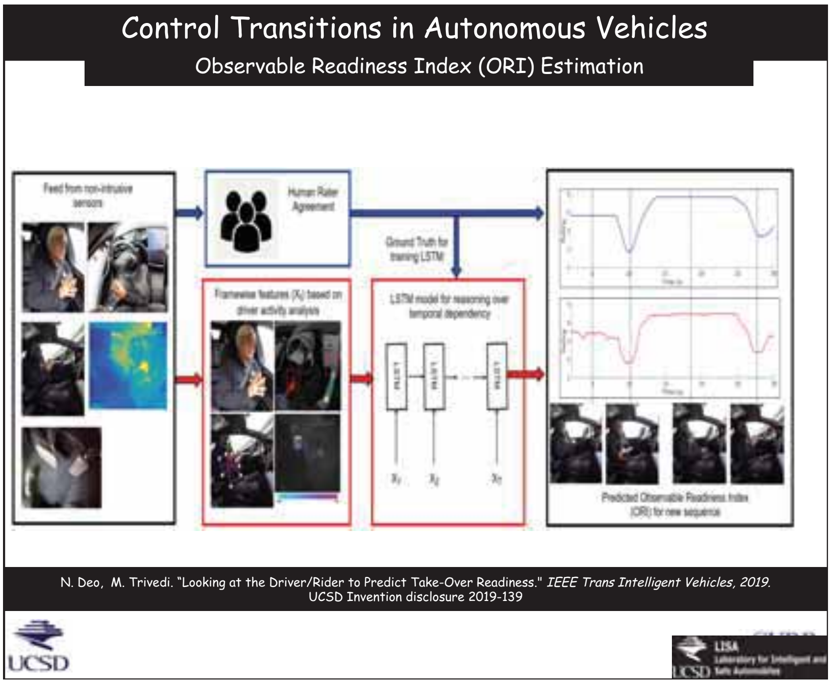
Looking at Hands

Classify Window at Wrist

- On Wheel
- Hover Wheel
- On Lap
- Radio
- In Air
- Cupholder

- Phone
- No Phone

Kevan Yuen and Mohan M. Trivedi, "Looking at Hands in Autonomous Vehicles:A ConvNet Approach using Part Affinity Fields," *IEEE Transactions on Intelligent Vehicles,* 2020.



Control Transitions in Autonomous Vehicles

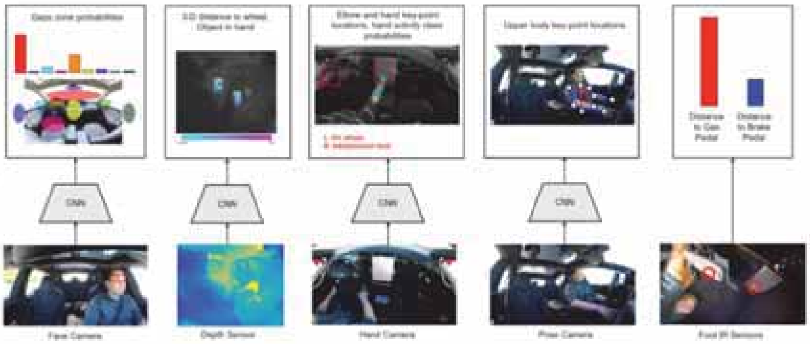
Observable Readiness Index (ORI) Estimation

Feed from non-intrusive sensors
Human Rater Agreement
Ground Truth for training LSTM
Framewise features (X_t) based on driver activity analysis
LSTM model for reasoning over temporal dependency
LSTM
LSTM
LSTM
x_1
x_2
x_T
Predicted Observable Readiness Index (ORI) for new sequence
UCSD
LISA Laboratory for Intelligent and Safe Automobiles

N. Deo, M. Trivedi. "Looking at the Driver/Rider to Predict Take-Over Readiness." *IEEE Trans Intelligent Vehicles, 2019.*
UCSD Invention disclosure 2019-139

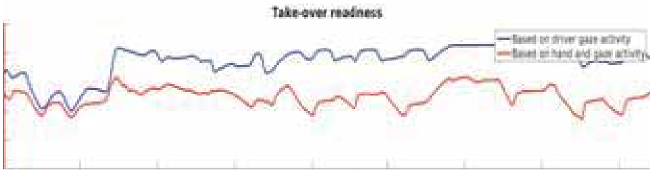
Control Transitions in Autonomous Vehicles

Observable Readiness Index (ORI) Estimation



N. Deo, M. Trivedi. "Looking at the Driver/Rider to Predict Take-Over Readiness." *IEEE Trans Intelligent Vehicles, 2019.*
UCSD Invention disclosure 2019-139

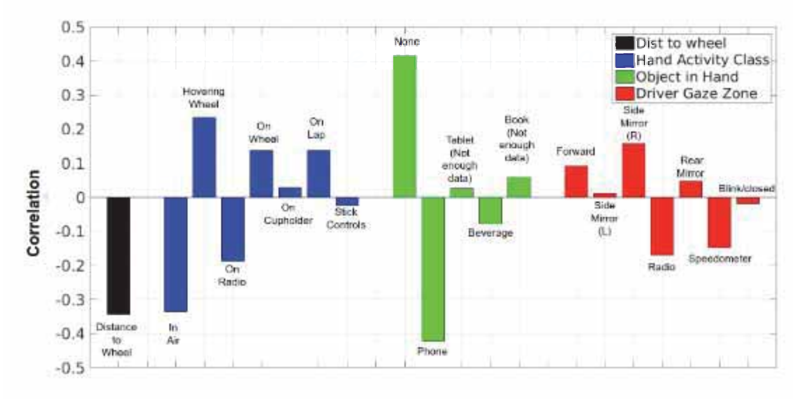
Observable Readiness Index (ORI) Estimation


Take-over readiness
Based on driver gaze activity
Based on hand and gaze activity

A Rangesh, N Deo, K Yuen, K Pirozhenko, M Trivedi, H Toyoda, P Gunaratne, "Exploring the Situational Awareness of Humans inside Autonomous Vehicles," *IEEE International Conference on Intelligent Transportation Systems* 2018

N. Deo, M. Trivedi. "Looking at the Driver/Rider to Predict Take-Over Readiness." *IEEE Trans Intelligent Vehicles, 2019.*
UCSD Invention disclosure 2019-139

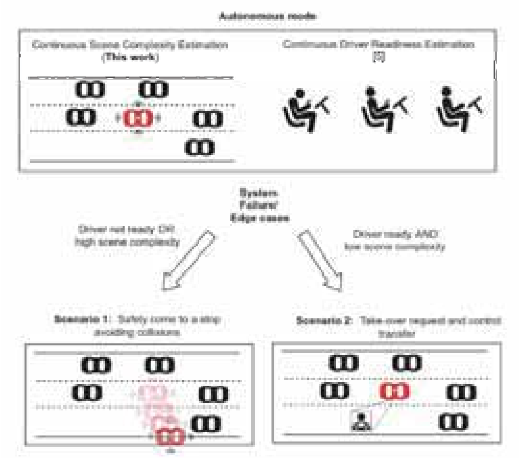
Rider Activity Correlations with Observable Readiness Index (ORI)



A Rangesh, N Deo, K Yuen, K Pirozhenko, M Trivedi, H Toyoda, P Gunaratne, "Exploring the Situational Awareness of Humans inside Autonomous Vehicles," *IEEE International Conference on Intelligent Transportation Systems* 2018

N. Deo, M. Trivedi. "Looking at the Driver/Rider to Predict Take-Over Readiness." *IEEE Trans Intelligent Vehicles, 2019.*
UCSD Invention disclosure 2019-139

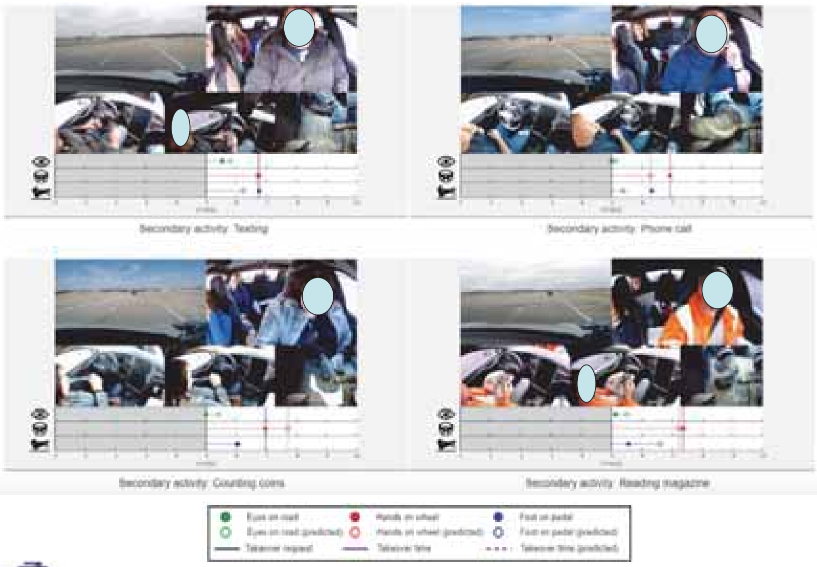
Exploring Control Transition and Driving Scene Complexity


Autonomous mode
Continuous Scene Complexity Estimation (This work)
Continuous Driver Readiness Estimation [5]
System Failure/ Edge cases
Driver not ready OR high scene complexity
Driver ready AND low scene complexity
Scenario 1: Safely come to a stop avoiding collisions
Scenario 2: Take-over request and control transfer

N Deo, N Meoli, A Rangesh M. Trivedi, "On Control Transitions in Autonomous Driving: A Framework and Analysis for Characterizing Scene Complexity," *ICCV Workshop on Autonomous Driving*, 2019.

Trivedi_IROS-PPNIV20

Take over Time Prediction



Takeover Time Prediction for Autonomous Vehicles: A Machine Learning Approach, UCSD Invention SD2021-070, 2020

Safe Autonomous Driving: Exciting journey continues !

Safe Autonomous Driving (AD) => Autonomous Vehicles + Humans

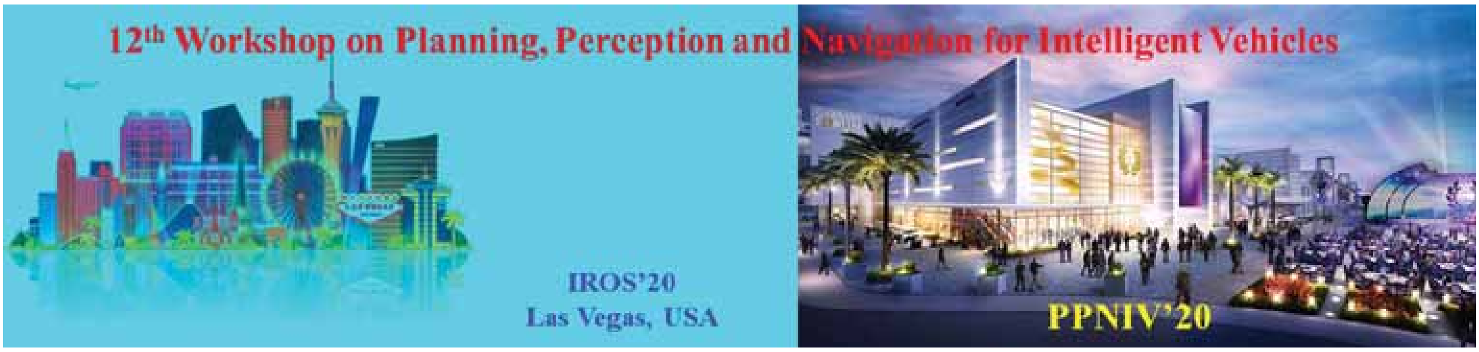
AD = Distributed Cognitive Systems: *Human-Vehicle Teams*

Research Explorations:

- Multiple Intelligent Agents,
- Holistic Situation Perception with Multimodal Sensors
- Understanding *Behavior and Interactions*
- *Predicting Intentions*
- Continuous Risk *Assessment*
- *Smooth/Safe Control Transitions, Fail-safe operation modes,*
- *Large Naturalistic Driving Studies and Sharable Datasets*
- *Evaluations, Metrics, and Benchmarks*
- *Reliability, Robustness and Scalability*

Thanks !

LISA Publications http://cvrr.ucsd.edu/publications/index.html


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

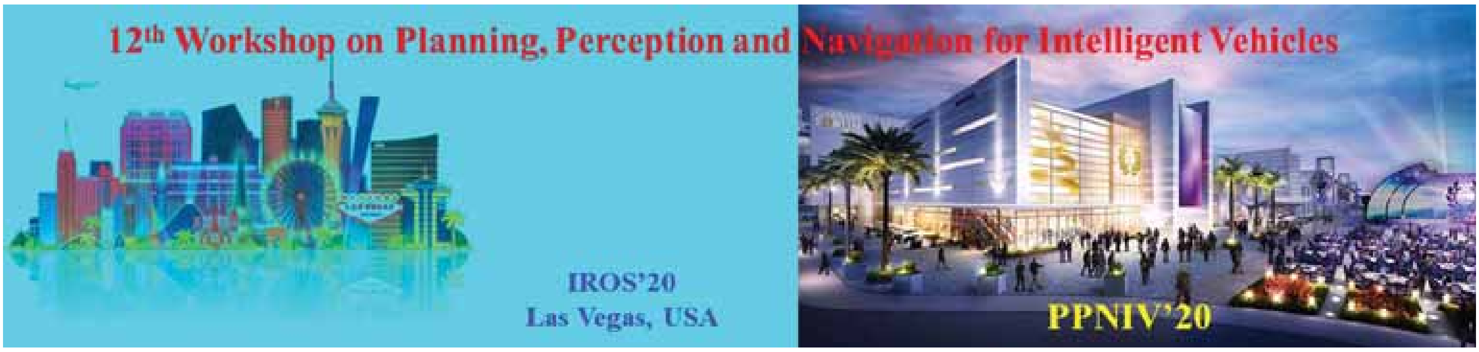
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Keynote speaker: **Evangelos Theodorou**
(Georgia Institute of Technology, USA)

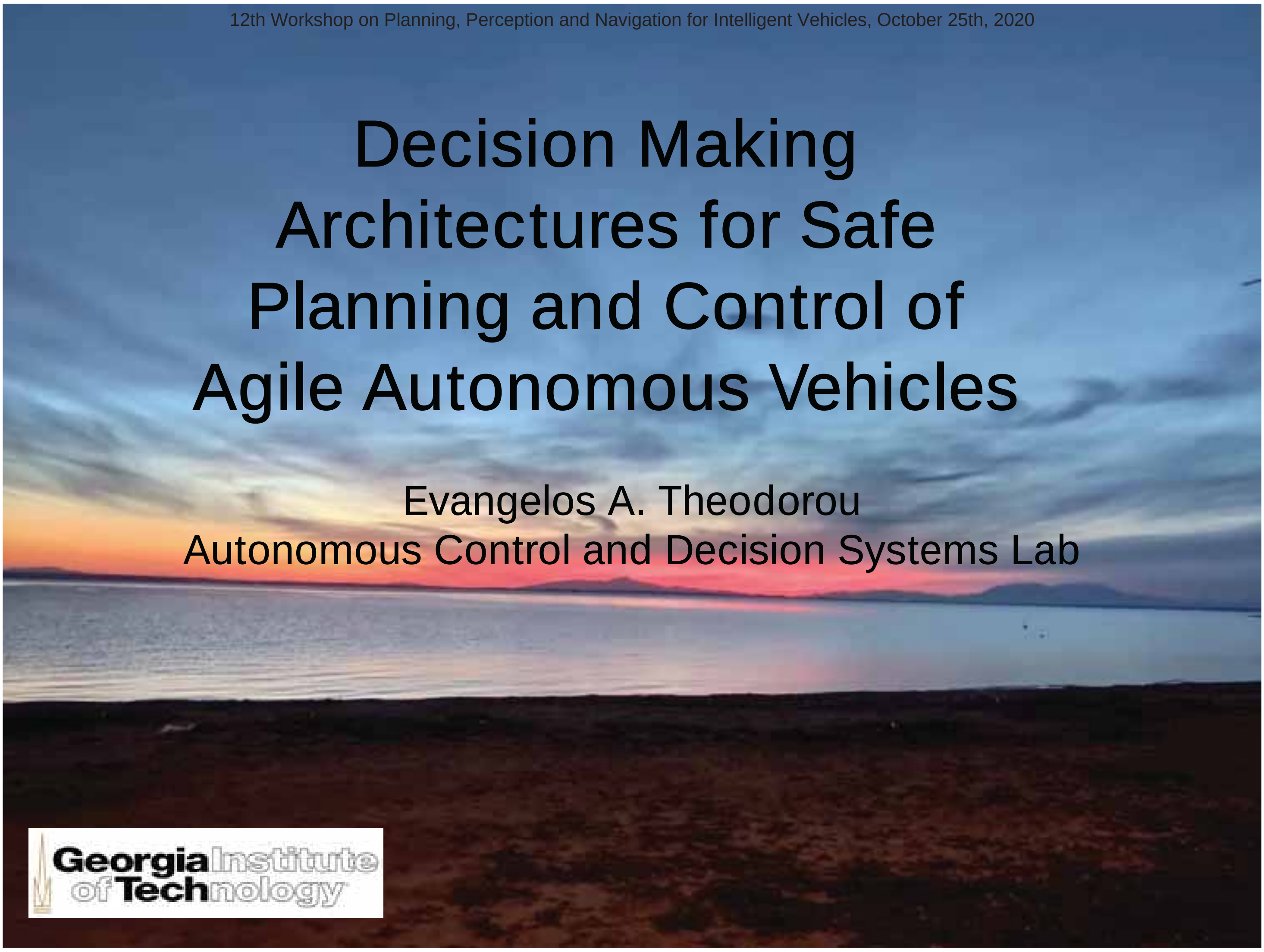
Decision Making Architectures for Safe Planning and Control of Agile Autonomous Vehicles

Abstract: In this talk I will present novel algorithms and decision-making architectures for safe planning and control of terrestrial and aerial vehicles operating in dynamic environments. These algorithms incorporate different representations of robustness for high speed navigation and bring together concepts from stochastic contraction theory, robust adaptive control, and dynamic stochastic optimization using augmented importance sampling techniques. I will present demonstrations on simulated and real robotic systems and discuss future research directions.

Biograpghy: Evangelos Theodorou is an Associate Professor with the School of Aerospace Engineering, Georgia Institute of Technology and is also the director of Autonomous Control and Decisions Systems (ACDS) laboratory. He is also affiliated with the Institute of Robotics and Intelligence Machines, and Center for Machine Learning Research at Georgia Tech. His interests are at the intersection stochastic control and optimization, machine learning, statistical physics and dynamic systems theory. Applications of his research include robotic and aerospace systems, applied physics, networked systems and bio-engineering.


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems



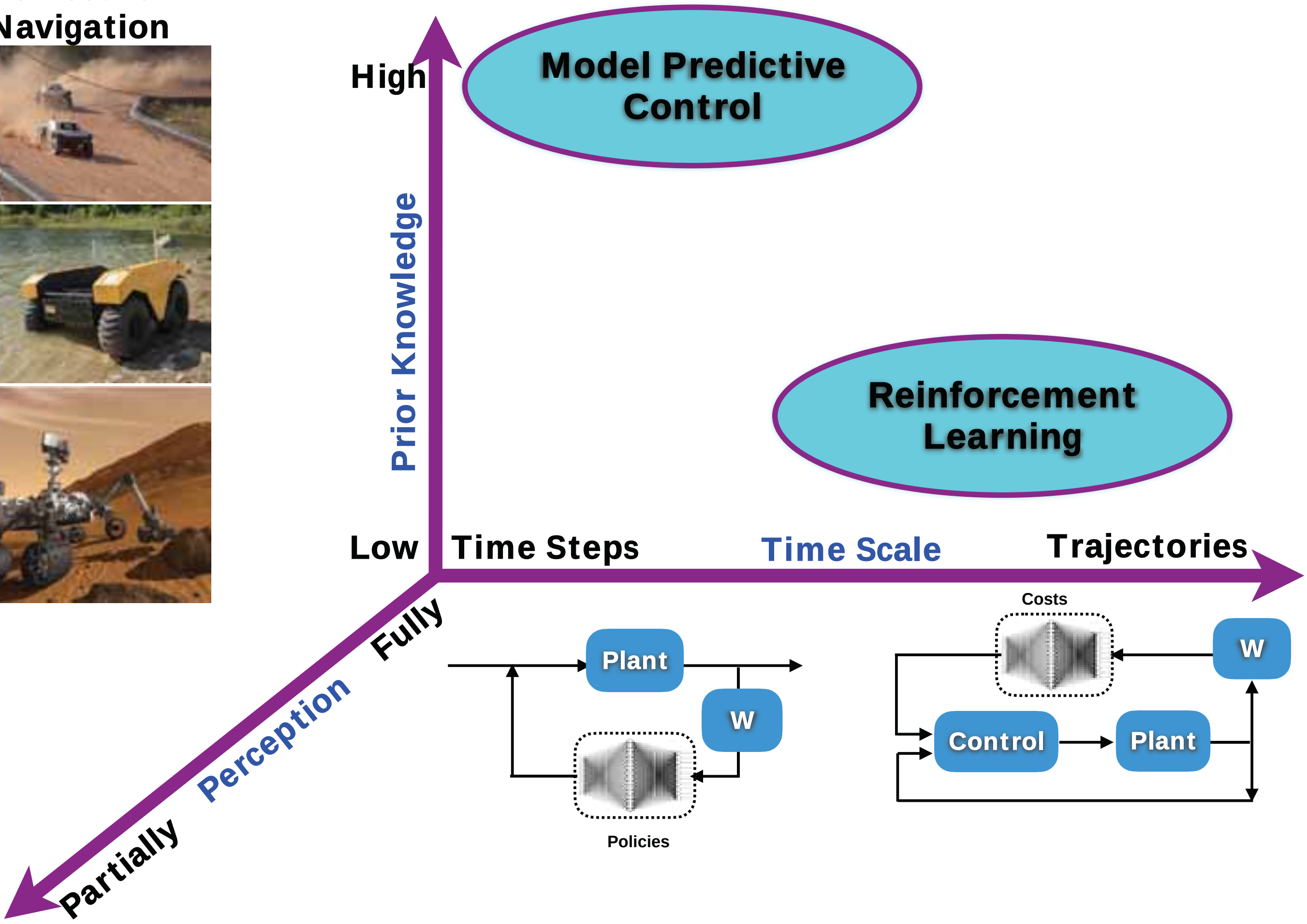
Decision Making Architectures for Safe Planning and Control of Agile Autonomous Vehicles

Evangelos A. Theodorou
Autonomous Control and Decision Systems Lab

Georgia Institute of Technology

Perceptual Decision Making

Terrestrial Navigation



High

Model Predictive Control

Prior Knowledge

Reinforcement Learning

Low

Time Steps

Time Scale

Trajectories

Fully

Perception

Partially

Plant

W

Policies

Costs

W

Control

Plant

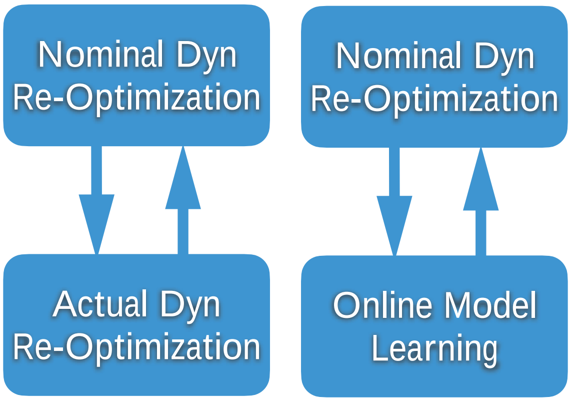
Outline

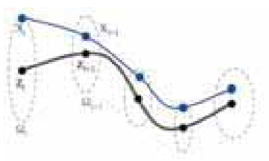
- ✦ Intro & Motivation
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusions and Future

What happens when uncertainty is not considered?

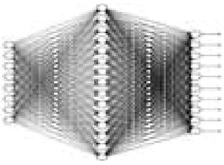
Information Processing Architectures

- ✦ How would you architect your stack?
- ✦ Where should learning be incorporated?
- ✦ What notions of robustness we have?

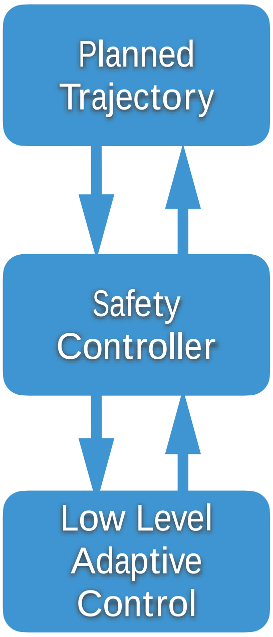

Nominal Dyn Re-Optimization
Actual Dyn Re-Optimization
Nominal Dyn Re-Optimization
Online Model Learning



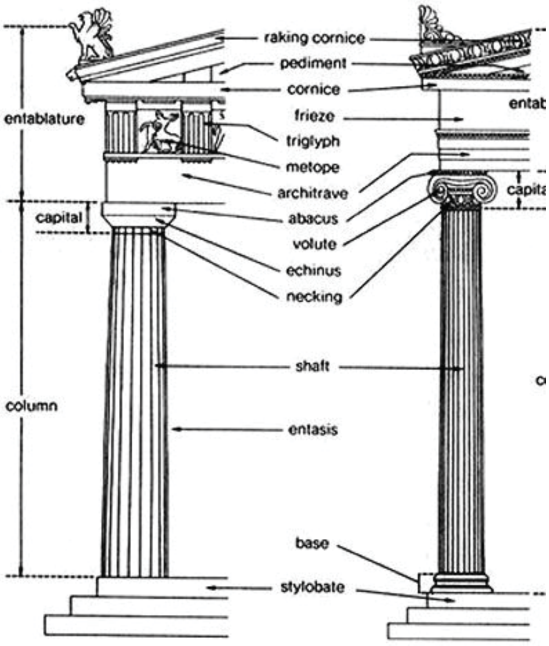
Tubes



Full Blown Model Learning

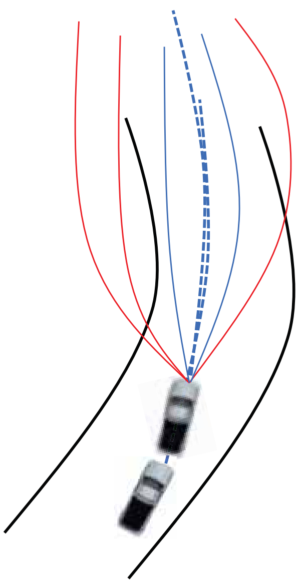


Adaptive + Predictive

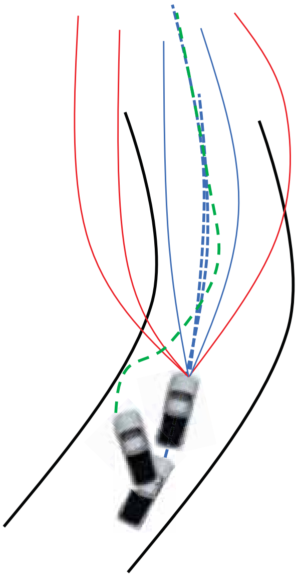


Model Predictive Path Integral (MPPI) Control

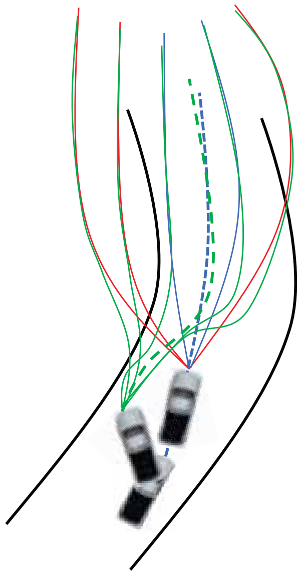
MPPI



Tube-MPPI



Robust MPPI



(-) Importance Sampler may get stuck to a local minima.

(-) Robustness issues when Large disturbances.

(-) Nominal State is chosen independent of Actual State.

(-) Importance Sampling is unaware of the underlying ancillary controller.

(+) Augmented Importance Sampling.

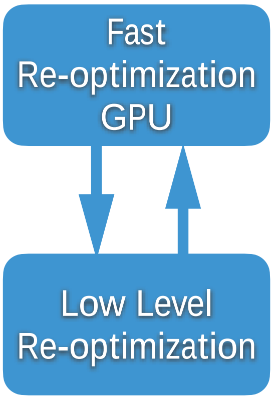
(+) Nice Trade-off between agility and robustness.

Robust MPPI

✓Performance near dynamic limits

✓Constraint satisfaction

✓Real-Time Performance



Fast Re-optimization GPU

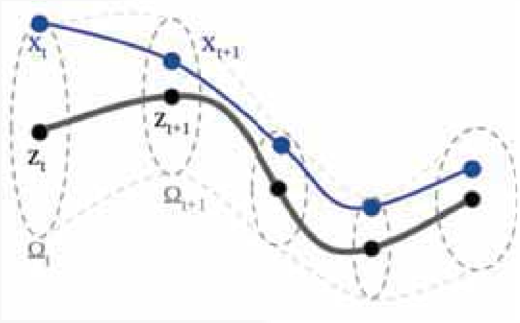
Low Level Re-optimization

NASA

Sandia National Laboratories

- ✦ Fast re-planning on GPU on nominal dynamics/Fast Tracking on a CPU
- ✦ Free Energy Diff < Levels Constraint Satisfaction + Tracking/Uncertainty + Sampling Error

Learning Deep Tubes for Robust MPC



$$z_{t+1} = f_z(z_t, v_t)$$

$$\omega_{t+1} = f_\omega(\omega_t, z_t, v_t, t)$$

$$P(d(x_t, z_t) \leq \omega_t) \geq \alpha, \quad \forall t \in \mathbb{N}$$

$$\pi(x,z) : \mathbb{X} \times \mathbb{Z} \to \mathbb{U}$$

$$d(x,z) = |P_{\mathbb{Z}}(x) - z| \in \mathbb{R}^{n_z}$$

$$\Omega_\omega(z) := \{x \in \mathbb{X} : d(x,z) \leq \omega\}$$

$$\Omega_\omega(z) := \{x \in \mathbb{X} : \|P_{\mathbb{Z}}(x) - z\|_\omega \leq 1\}$$

$$\min_{v_{\cdot|t} \in \mathbb{V}} J_T(v_{\cdot|t}, z_{\cdot|t}, \omega_{\cdot|t})$$

$$s.t. \forall k = 0, \cdots, T :$$

$$z_{k+1|t} = f_z(z_{k|t}, v_{k|t})$$

$$\omega_{k+1|t} = f_\omega^\theta(\omega_t, z_t, v_t, t)$$

$$\omega_{0|t} = d(x_t, z_{0|t})$$

$$z_{T|t} = f_z(z_{T|t}, v_{T|t})$$

$$\omega_{T|t} \geq f_\omega^\theta(\omega_{T|t}, z_{T|t}, v_{T|t}, T)$$

$$\Omega_{\omega_{k|t}}(z_{k|t}) \subseteq \mathcal{C}$$

Theorem III.1. *Suppose that the MPC problem (13) is feasible at $t=0$. Then the problem is feasible for all $t > 0 \in \mathbb{N}$ and at each timestep the constraints are satisfied with probability α.*

Georgia Tech

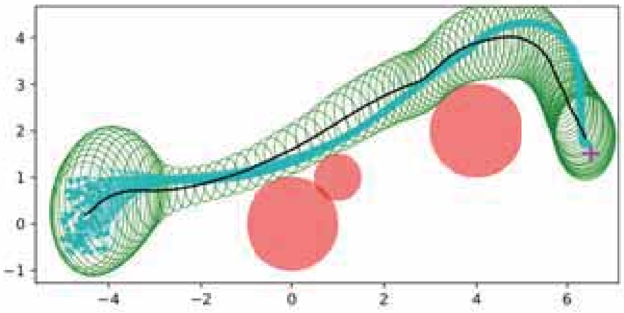
Learning Deep Tubes for Robust MPC

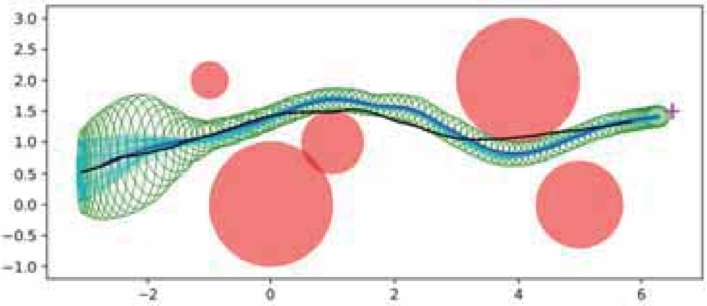
$$\dot{p}_x = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & -k_f\end{bmatrix} p_x + \begin{bmatrix}0\\0\\1\end{bmatrix} u_x + \begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix} w$$

$$\dot{z}_x = \begin{bmatrix}0 & 1\\ 0 & -k_f^z\end{bmatrix} z_x + \begin{bmatrix}0\\1\end{bmatrix} v_x \quad w \sim \mathcal{N}(0, \epsilon I_{2\times 2})$$

$$e_x = k_e(z_x - p_x)$$

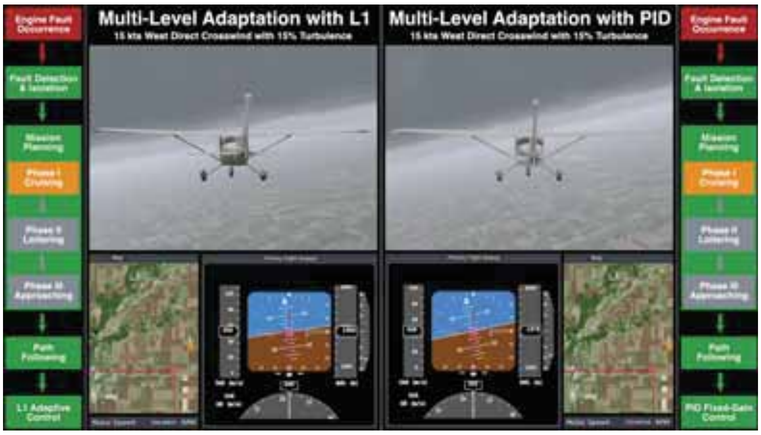
$$\pi_x(p_x, z_x) = k_p(e_x - \dot{p}_x + \dot{z}_x) + k_d(-\ddot{p}_x)$$

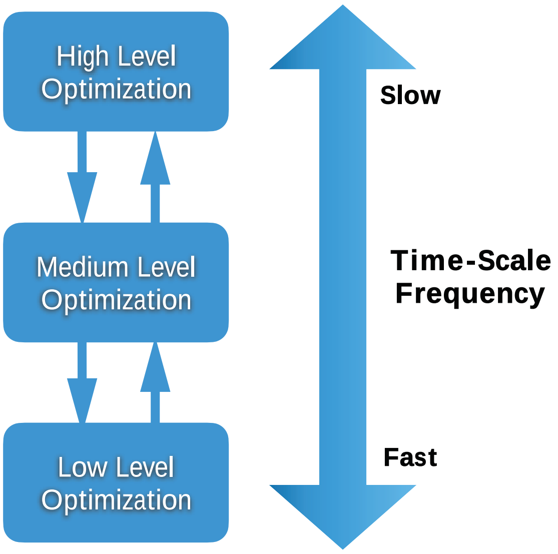




Sully Miracle on the Hudson

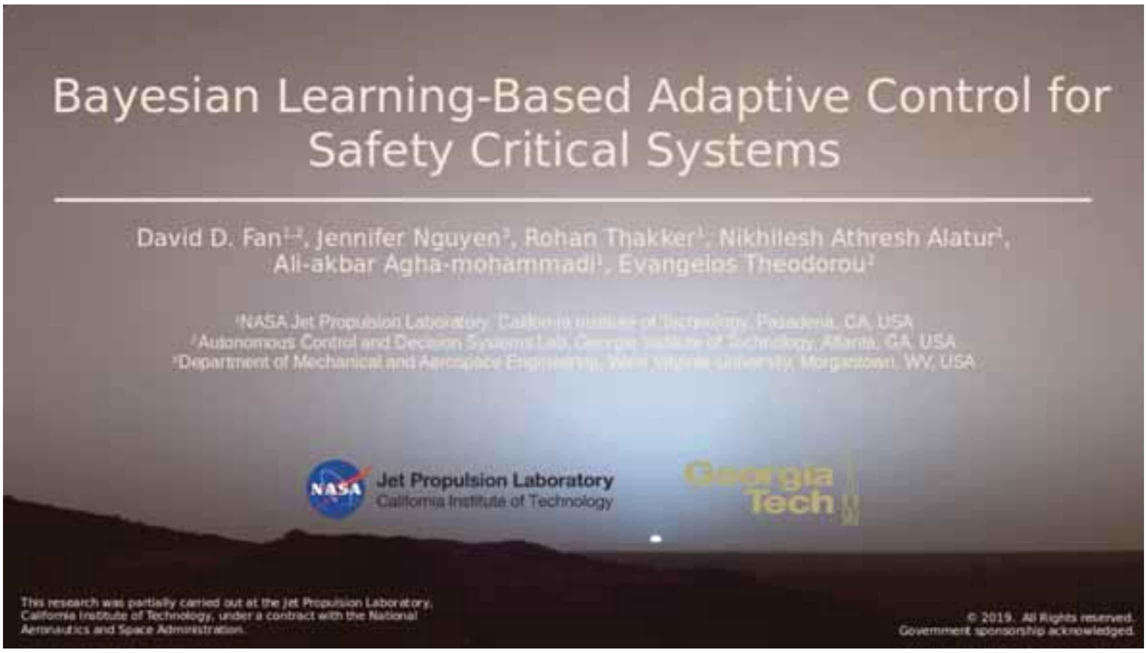
- Airbus 320 lost both engines shortly after takeoff due to bird strike.


Multi-Level Adaptation with L1
15 kts West Direct Crosswind with 15% Turbulence
Multi-Level Adaptation with PID
15 kts West Direct Crosswind with 15% Turbulence
Engine Fault Occurrence
Fault Detection & Isolation
Mission Planning
Phase I Cruising
Phase II Loitering
Phase III Approaching
Path Following
L1 Adaptive Control
PID Fixed-Gain Control



Courtesy: NASA Langley Aerodrome

Adaptation and Online Learning



Bayesian Learning-Based Adaptive Control for Safety Critical Systems

David D. Fan[1,2], Jennifer Nguyen[3], Rohan Thakker[1], Nikhilesh Athresh Alatur[1], Ali-akbar Agha-mohammadi[1], Evangelos Theodorou[2]

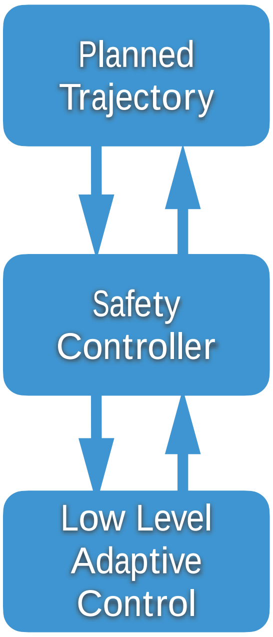
[1]NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
[2]Autonomous Control and Decision Systems Lab, Georgia Institute of Technology, Atlanta, GA, USA
[3]Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV, USA

Jet Propulsion Laboratory
California Institute of Technology

Georgia Tech

This research was partially carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

© 2019. All Rights reserved.
Government sponsorship acknowledged.



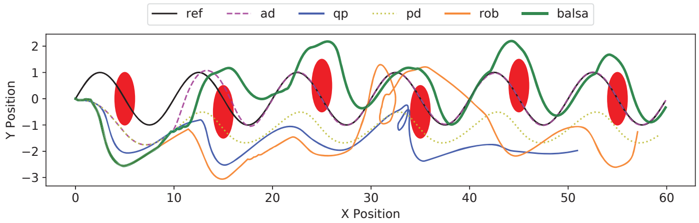
Planned Trajectory

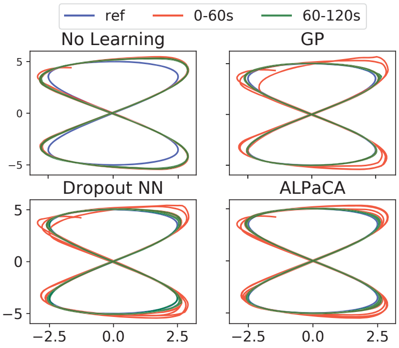
Safety Controller

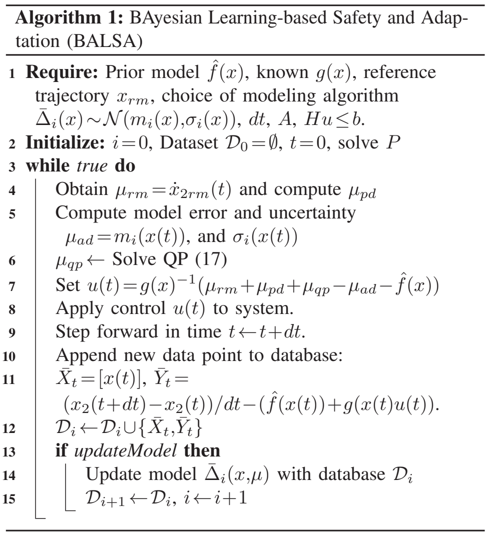
Low Level Adaptive Control

- ✦ Stochastic Control Barrier Functions
- ✦ Stochastic Control Lyapunov Functions
- ✦ Bayesian Neural Networks

How do we bring adaptation?

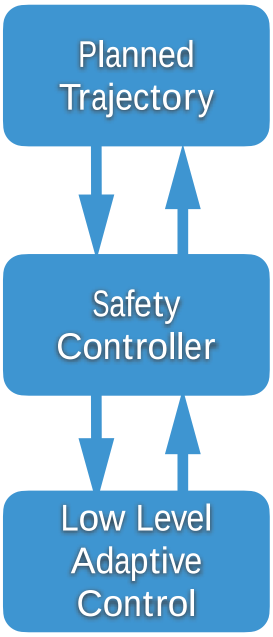






Algorithm 1: BAyesian Learning-based Safety and Adaptation (BALSA)

```
Require: Prior model $\hat{f}(x)$, known $g(x)$, reference
  trajectory $x_{rm}$, choice of modeling algorithm
  $\bar{\Delta}_i(x) \sim \mathcal{N}(m_i(x), \sigma_i(x))$, $dt$, $A$, $Hu \leq b$.
Initialize: $i=0$, Dataset $\mathcal{D}_0 = \emptyset$, $t=0$, solve $P$
while true do
    Obtain $\mu_{rm} = \dot{x}_{2rm}(t)$ and compute $\mu_{pd}$
    Compute model error and uncertainty
      $\mu_{ad} = m_i(x(t))$, and $\sigma_i(x(t))$
    $\mu_{qp} \leftarrow$ Solve QP (17)
    Set $u(t) = g(x)^{-1}(\mu_{rm} + \mu_{pd} + \mu_{qp} - \mu_{ad} - \hat{f}(x))$
    Apply control $u(t)$ to system.
    Step forward in time $t \leftarrow t + dt$.
    Append new data point to database:
    $\bar{X}_t = [x(t)]$, $\bar{Y}_t =$
      $(x_2(t+dt) - x_2(t))/dt - (\hat{f}(x(t)) + g(x(t)u(t))$.
    $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{\bar{X}_t, \bar{Y}_t\}$
    if updateModel then
        Update model $\bar{\Delta}_i(x, \mu)$ with database $\mathcal{D}_i$
        $\mathcal{D}_{i+1} \leftarrow \mathcal{D}_i$, $i \leftarrow i+1$
```



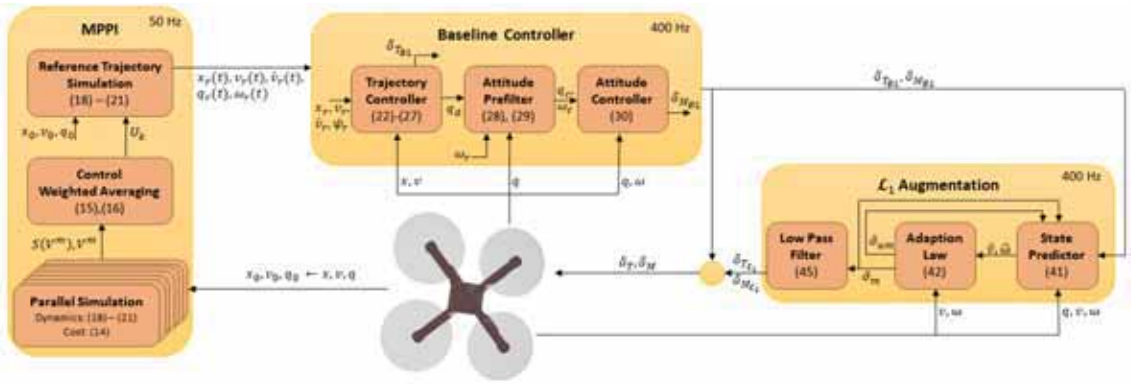
Planned Trajectory

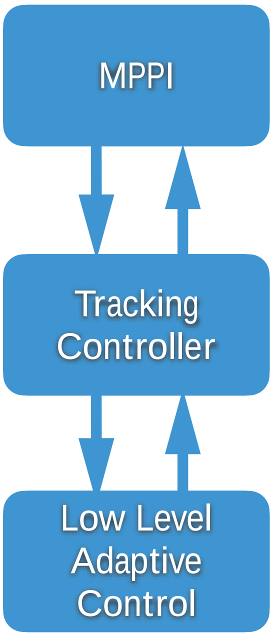
Safety Controller

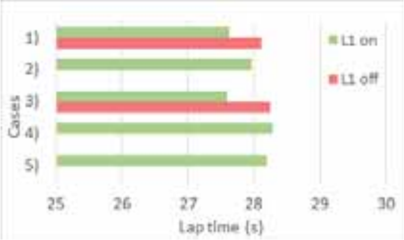
Low Level Adaptive Control

- ✦ Stochastic Control Barrier Functions
- ✦ Stochastic Control Lyapunov Functions
- ✦ Bayesian Neural Networks

Adaptation and Online Learning


MPPI
50 Hz
Reference Trajectory Simulation (18) – (21)
x_0, v_0, q_0
U_k
Control Weighted Averaging (15),(16)
$S(V^m), V^m$
Parallel Simulation
Dynamics: (18) – (21)
Cost: (14)
$x_r(t), v_r(t), \dot{v}_r(t), q_r(t), \omega_r(t)$
Baseline Controller
400 Hz
$\delta_{T_{BL}}$
Trajectory Controller (22)-(27)
$x_r, v_r, \dot{v}_r, \psi_r$
q_d
Attitude Prefilter (28), (29)
ω_r
q_r, ω_r
Attitude Controller (30)
$\delta_{M_{BL}}$
x, v
q
q, ω
$\delta_{T_{BL}}, \delta_{M_{BL}}$
$\mathcal{L}_1$ Augmentation
400 Hz
Low Pass Filter (45)
Adaption Law (42)
State Predictor (41)
$\hat{\sigma}_{um}$
$\hat{\sigma}_m$
$\tilde{v}, \tilde{\Omega}$
$\delta_{T_{L_1}}, \delta_{M_{L_1}}$
δ_T, δ_M
$x_0, v_0, q_0 \leftarrow x, v, q$
v, ω
q, v, ω





1) known dynamics model (since drag is not modeled in the nominal dynamics, some drag compensation is expected with $\mathcal{L}_1$ augmentation);
2) mass increase by 50%;
3) moment of inertia increase by 100% in all axes;
4) constant nose-up pitching moment disturbance of 0.1 Nm (equivalent to center of gravity offset);
5) reduction in motor thrust control power by 40% (reduction in both $\bar{T}_{\delta_T}$ and $\bar{\mathbf{M}}_{\delta_M}$).

Case	$\mathcal{L}_1$ off	$\mathcal{L}_1$ on
1)	✓	✓
2)	✗	✓
3)	✓	✓
4)	✗	✓
5)	✗	✓

Adaptation and Online Learning

1) known dynamics model (since drag is not modeled in the nominal dynamics, some drag compensation is expected with $\mathcal{L}_1$ augmentation);
2) mass increase by 50%;
3) moment of inertia increase by 100% in all axes;
4) constant nose-up pitching moment disturbance of 0.1 Nm (equivalent to center of gravity offset);
5) reduction in motor thrust control power by 40% (reduction in both $\bar{T}_{\delta_T}$ and $\bar{\mathbf{M}}_{\delta_M}$).

Case	$\mathcal{L}_1$ off	$\mathcal{L}_1$ on
1)	✓	✓
2)	✗	✓
3)	✓	✓
4)	✗	✓
5)	✗	✓

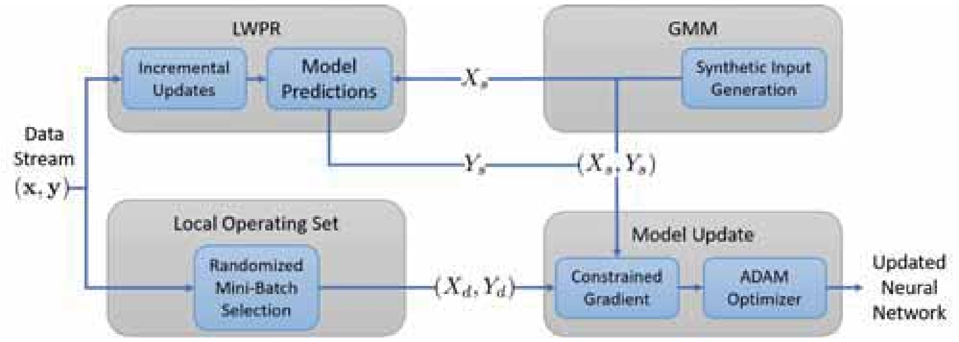
Adaptation and Online Learning

System ID Distribution: $\mathbf{x} \sim \mathcal{P}_{ID}(\mathcal{X})$ **Local Distribution:** $\mathbf{x} \sim \mathcal{P}_{L}(\mathcal{X})$

Update Scheme: $$\theta_{i+1} = \theta_i - \gamma G_L(\theta_i) \longrightarrow \theta_{i+1} = \theta_i - \gamma(G_L(\theta_i) + G_{ID}(\theta_i))$$

Proposed Scheme: $$\theta_{i+1} = \theta_i - \gamma\left(\alpha G_L(\theta_i) + G_{ID}(\theta_i)\right)$$
$$\alpha = \max_{a\in[0,1]} \quad s.t \quad \langle a G_L(\theta_i) + G_{ID}(\theta_i), G_{ID}(\theta_i)\rangle \geq 0$$

LWPR: $$\mathbf{y} = \sum_{i=1}^{L} w_i \cdot f_i(\mathbf{x} - \mathbf{c}_i), \quad w_i = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{c}_I)^{\mathrm{T}} D_i (\mathbf{x}-\mathbf{c}_i)\right)}{\sum_{j=1}^{L} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{c}_j)^{\mathrm{T}} D_j (\mathbf{x}-\mathbf{c}_j)\right)}$$



G. Williams et all, arXiv:1905.05162, Submitted

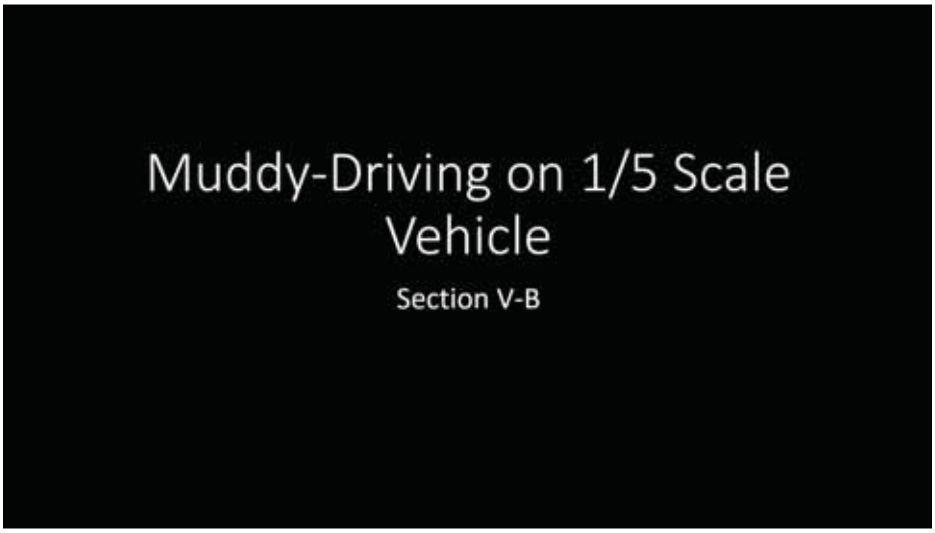
Adaptive Model Predictive Control

Computation

	Size	FLOPs/Prediction
LWPR	5,645 (Receptive Fields)	$> 141,125$
Neural Network	1,412 (Weights and Biases)	$2,688$

Performance

	Base	SGD	LW-PR^2	LWPR
Roll Rate	0.01	0.01	0.01	0.01
Long. Acc.	2.73	2.28	2.30	**2.06**
Lat. Acc.	1.71	1.29	**1.24**	1.28
Head. Acc.	8.28	**4.48**	4.87	4.54
Total MSE	3.18	2.10	2.11	**1.97**
Active MSE	N/A	N/A	2.54	N/A



Muddy-Driving on 1/5 Scale Vehicle

Section V-B

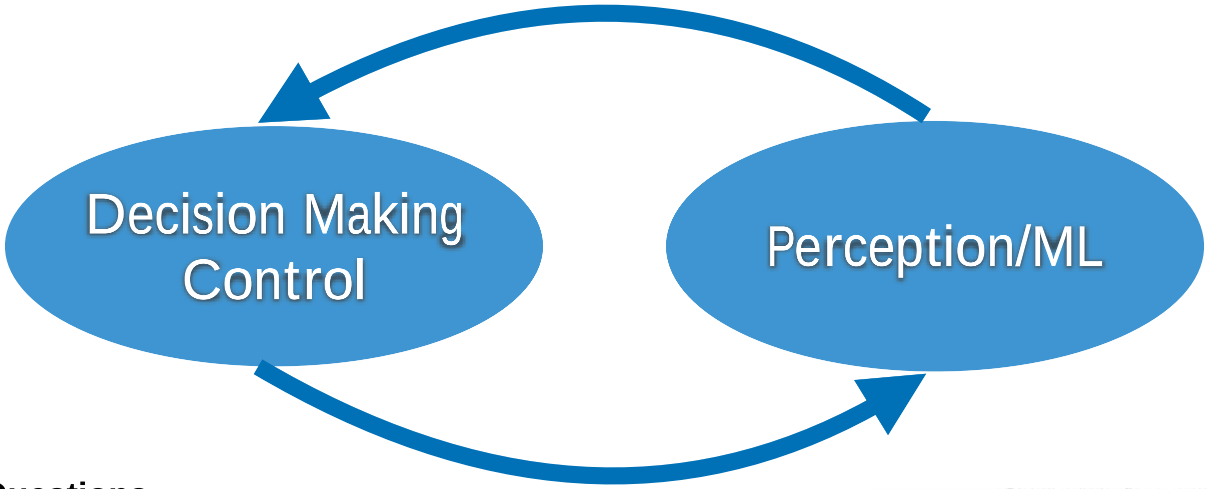
Outline

- ✦ Motivation & Intro
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusiona and Future

Outline

- ✦ Motivation & Intro
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusion and Future

Information Processing Architecture (IPA) for Perceptual control

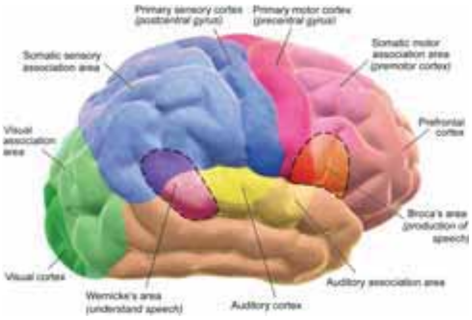


Questions:

What is the optimal **IPA** for perceptual control?

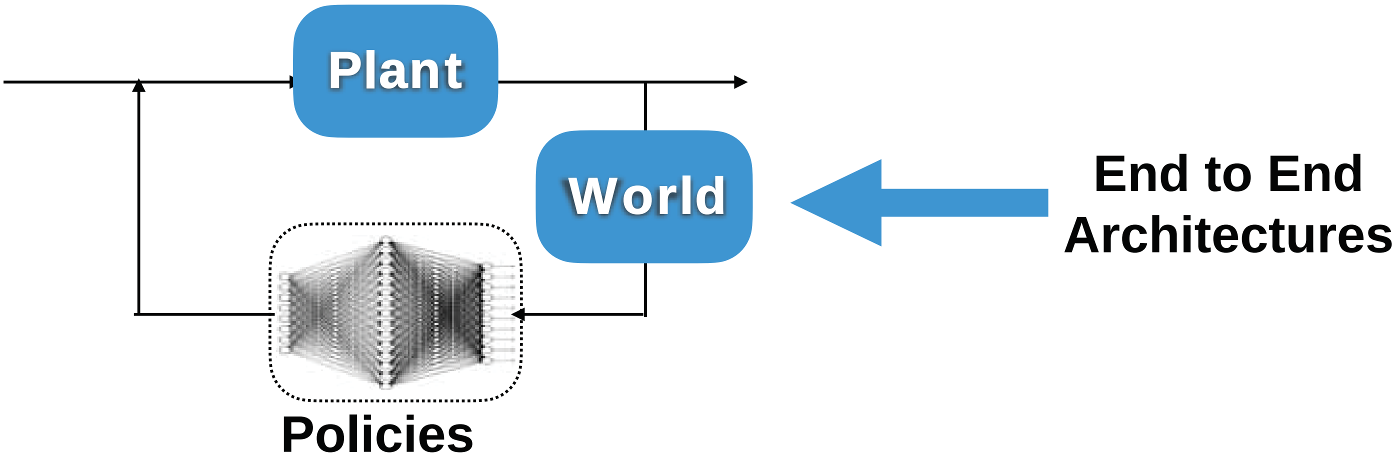
Is the design of **IPA** imposed by the nature of the data?

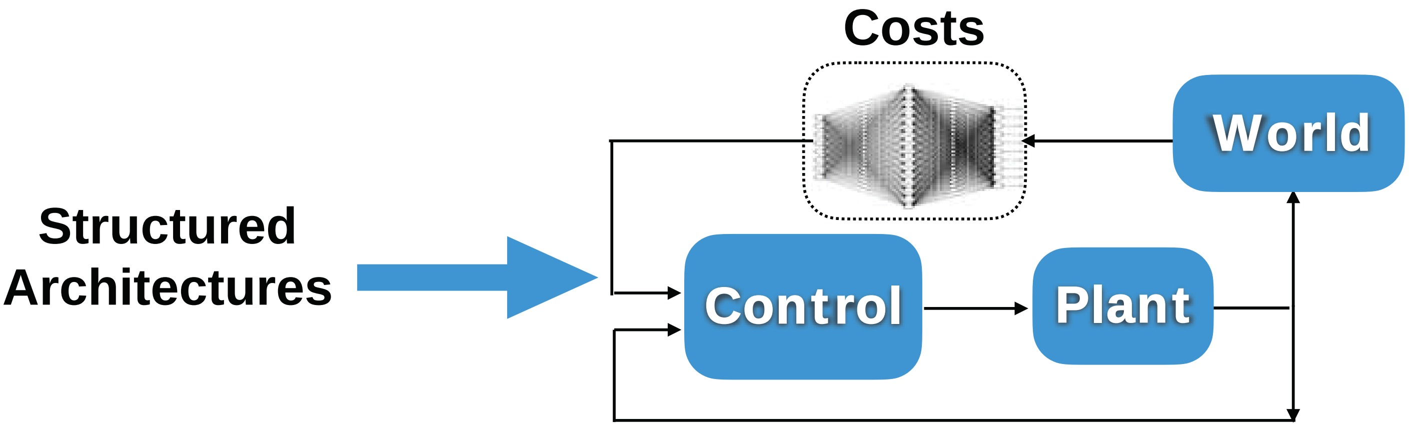
Do we have any priors for designing **IPAs**?


Primary sensory cortex (postcentral gyrus)
Primary motor cortex (precentral gyrus)
Somatic sensory association area
Somatic motor association area (premotor cortex)
Visual association area
Prefrontal cortex
Broca's area (production of speech)
Visual cortex
Wernicke's area (understand speech)
Auditory cortex
Auditory association area

How important is the structure of **IPAs** for **safety** in AI?

Information Processing Architectures : IPA

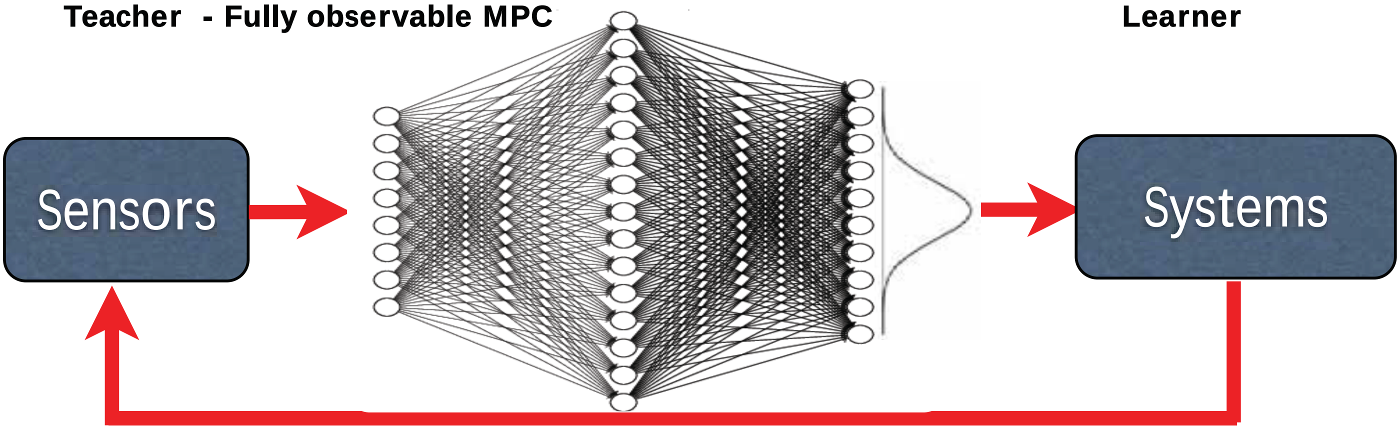




IPA-I

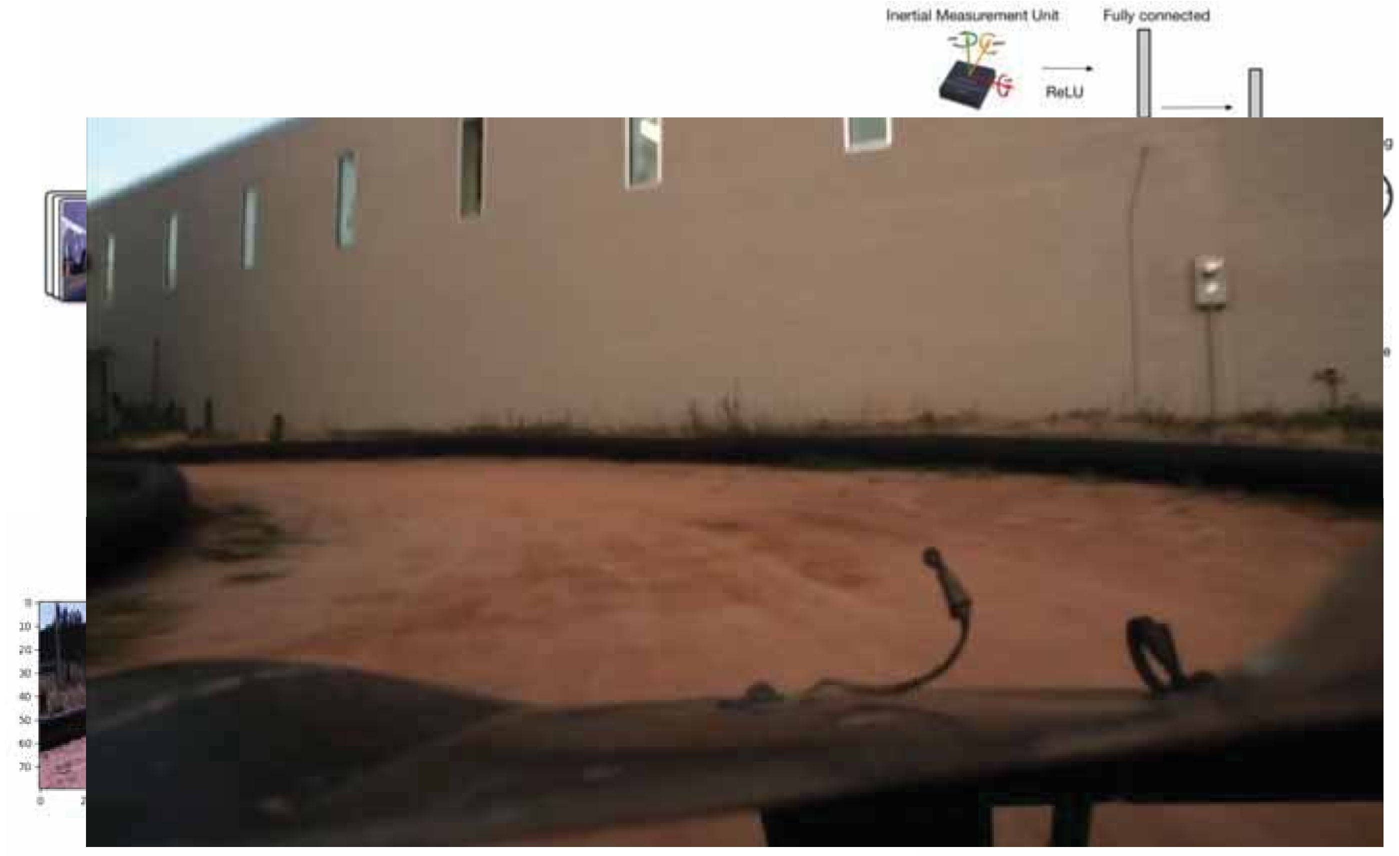
Teacher - Fully observable MPC

Learner



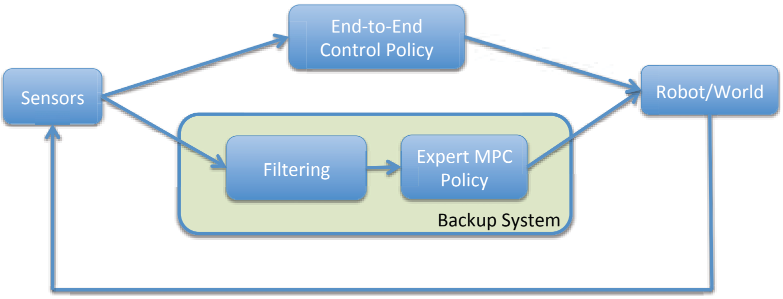
Y. Pan et all RSS 2018.

IPA-I



Y. Pan et all RSS 2018.

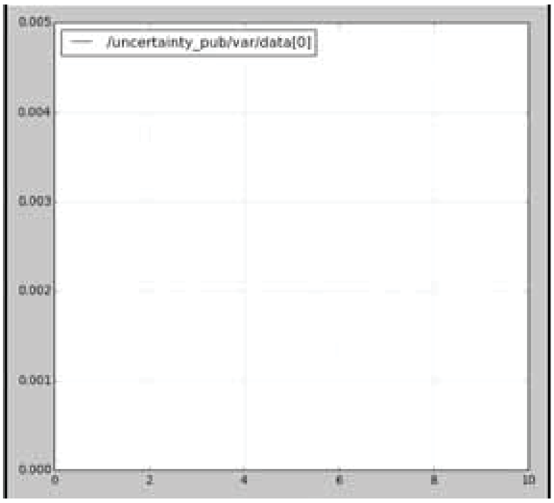
IPA-I & Uncertainty Quantification

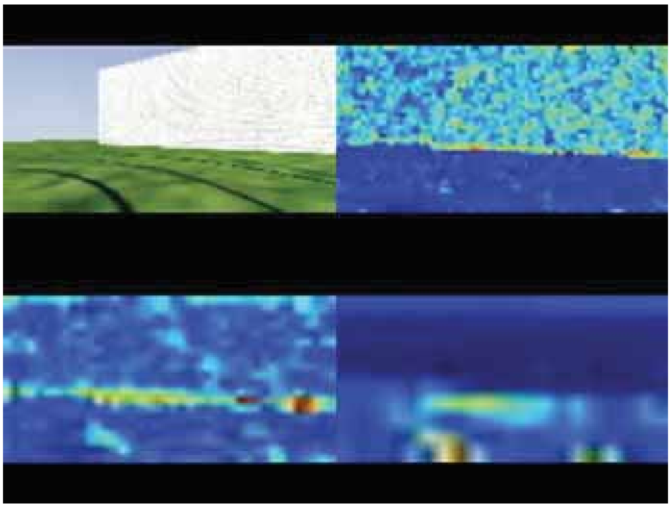


Types of Uncertainty in ML Models

Aleatoric - Incomplete data

Epistemic- Incomplete knowledge of the environment.



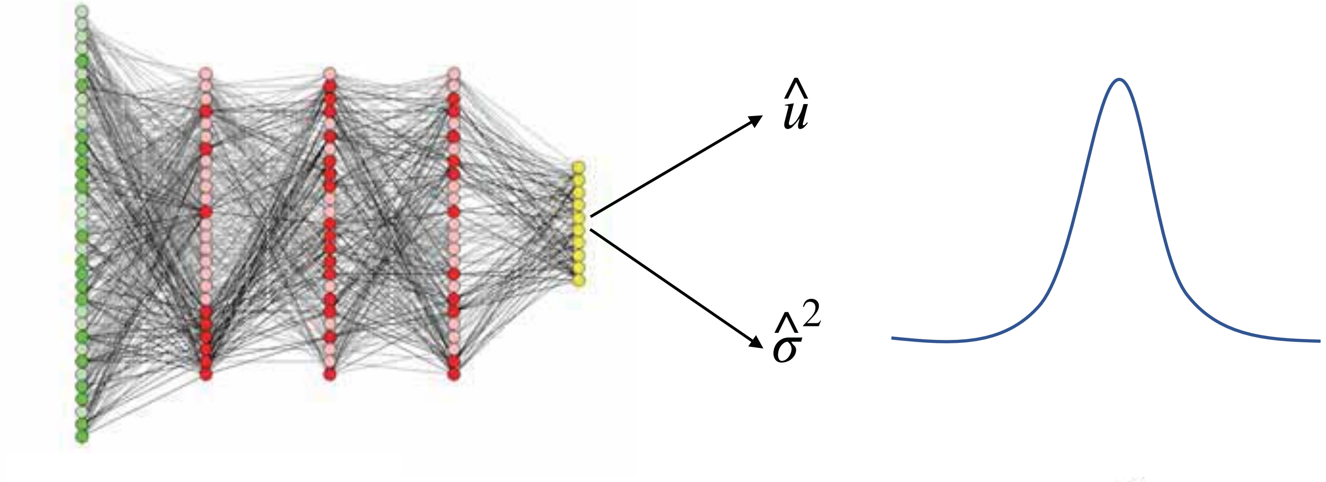


K. Lee et all ICRA 2019.

IPA-I & Uncertainty Quantification

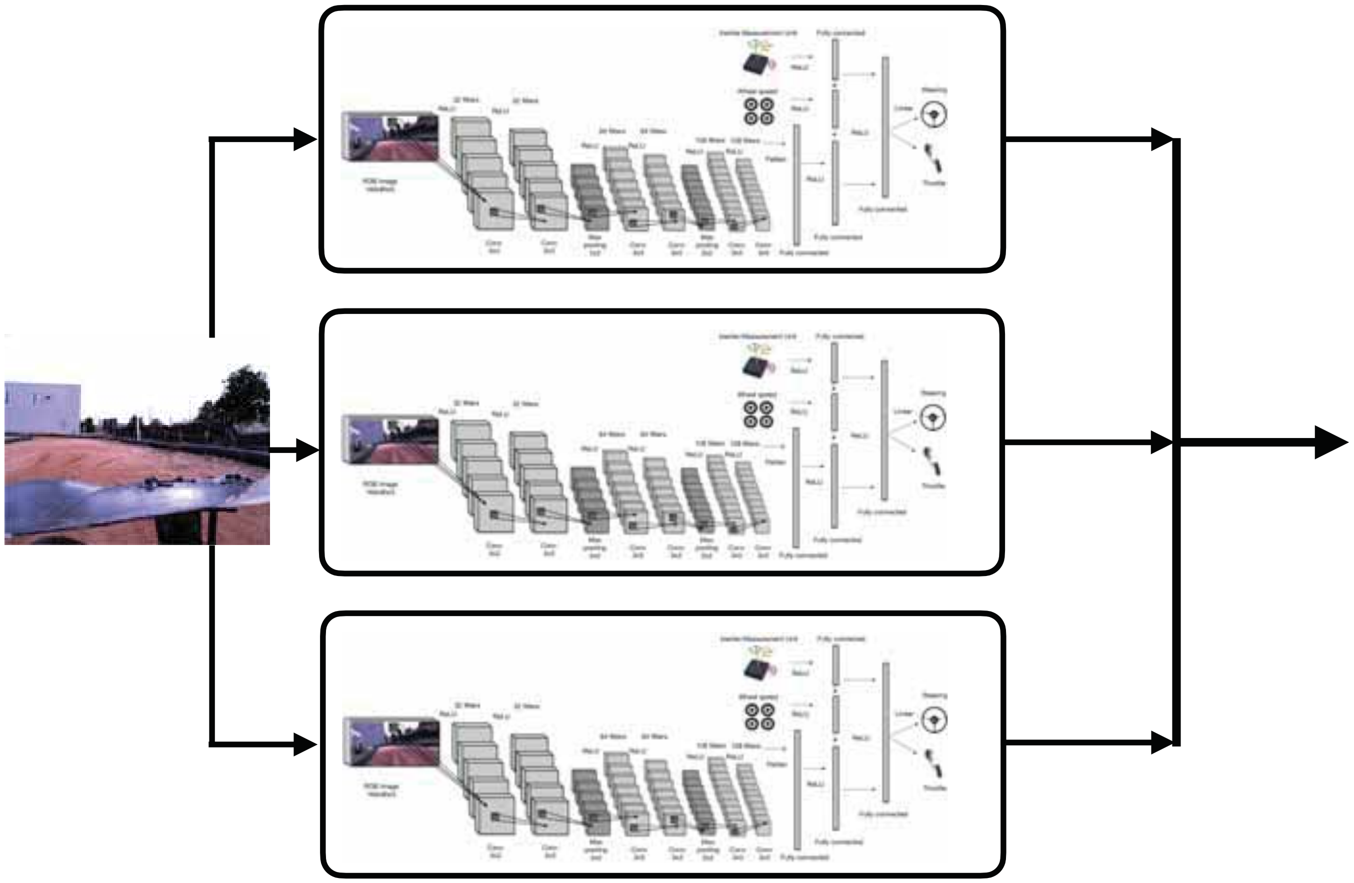
At Training Time Minimize the Loss: $\mathcal{L}(\pi) = \frac{1}{2\hat{\sigma}^2}||u^* - \hat{u}||^2 + \frac{1}{2}\log(\hat{\sigma}^2)$

At Test Time Sample the structure of the Network:

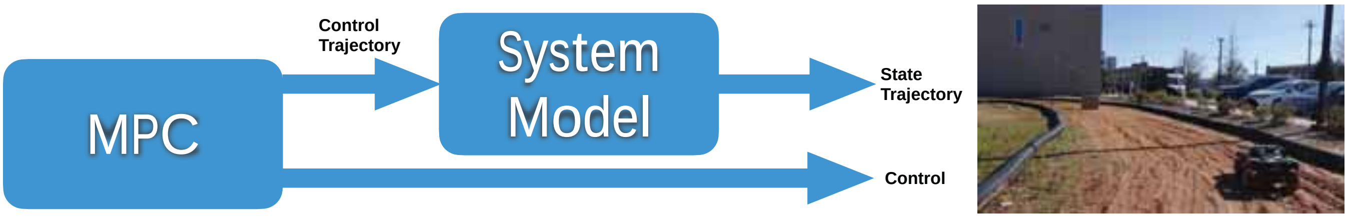


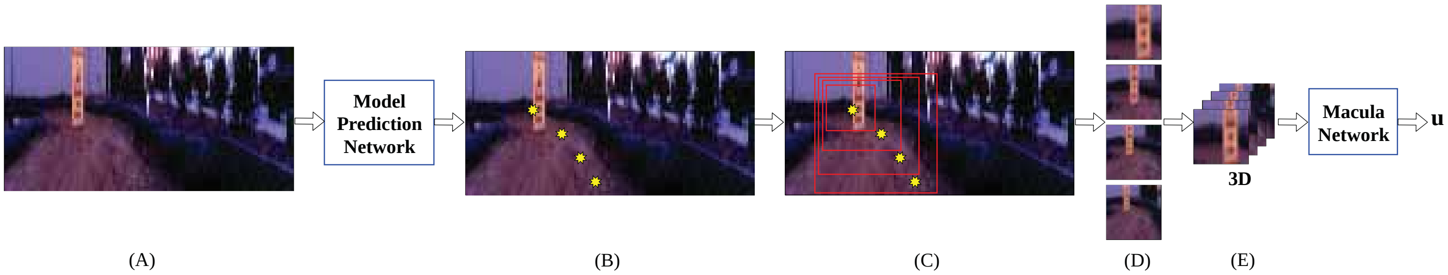
Total Uncertainty: $\sigma_i^2 = Var(u_i) \approx \underbrace{\frac{1}{K}\sum_{k=1}^{K} \hat{u}_{i_k}^2 - \left(\frac{1}{K}\sum_{k=1}^{K} \hat{u}_{i_k}\right)^2}_{epistemic} + \underbrace{\frac{1}{K}\sum_{k=1}^{K} \hat{\sigma}_{i_k}^2}_{aleatoric}$

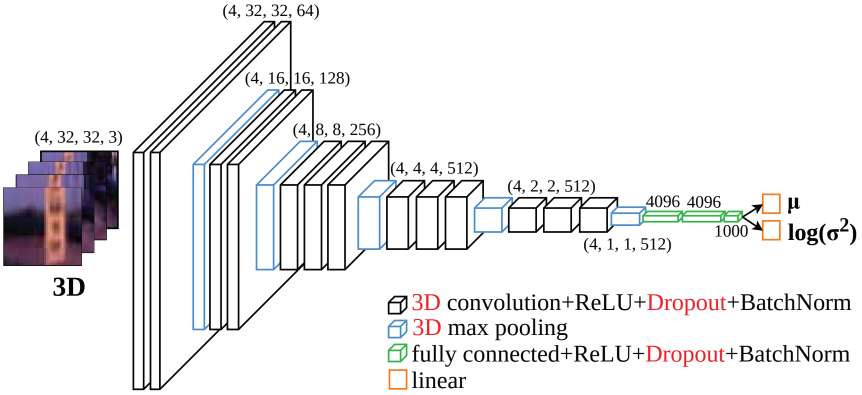
Uncertainty Quantification & Redundancy



IPA-II: The Macula-Net

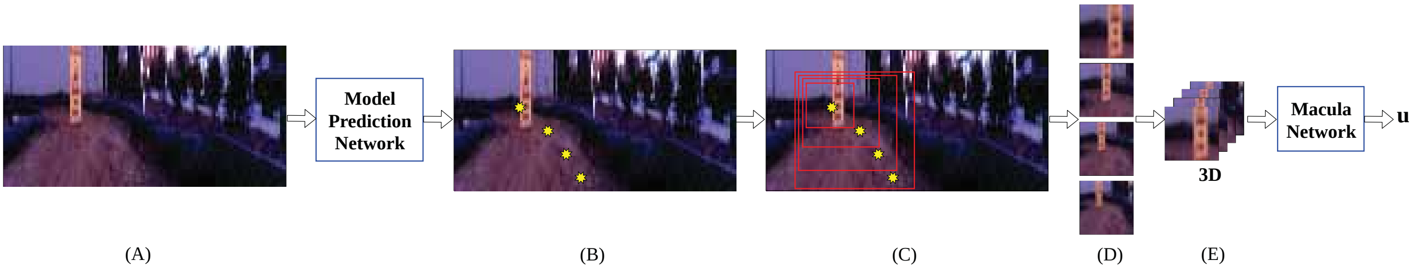






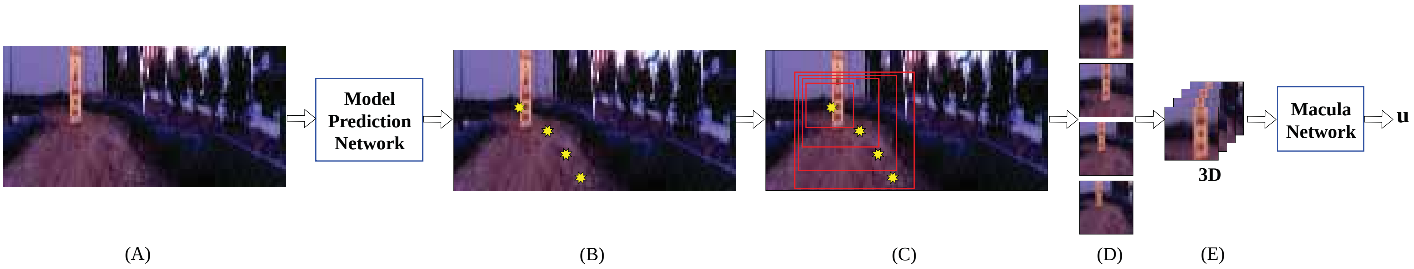
K. Lee et all, arXiv:1904.11898, Submitted

IPA-II: The Macula-Net

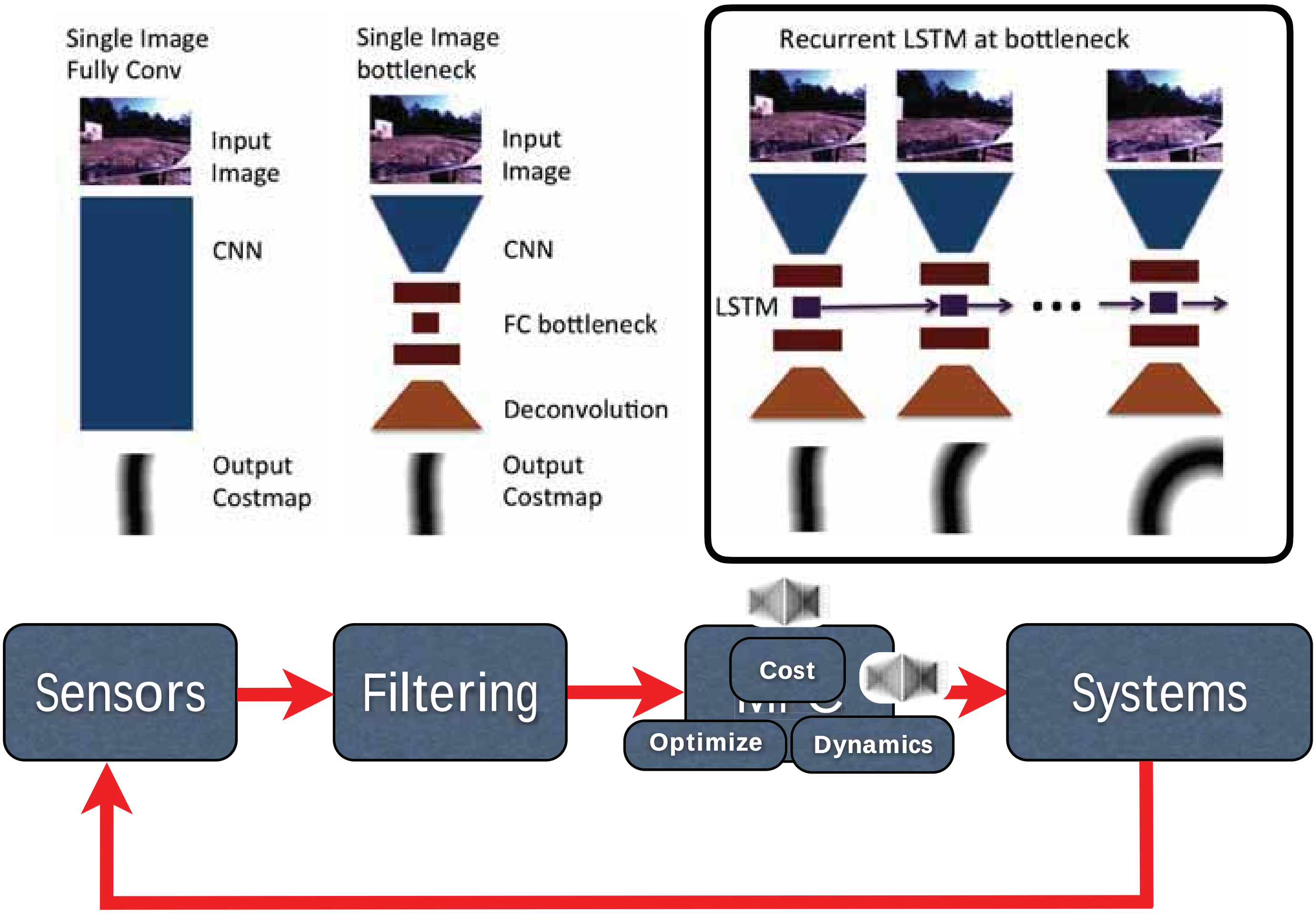


New Objects	DropoutVGG [9]	PAPC [Ours]
	Min: 0.37 m Avg: 0.39 m Max: 0.42 m	Min: **4.28** m Avg: **6.87** m Max: **9.22** m
	Min: 2.20 m Avg: 2.54 m Max: 2.86 m	Min: **4.81** m Avg: **5.48** m Max: **6.25** m
	Min: 0.00 m Avg: 0.00 m Max: 0.00 m	Min: **6.80** m Avg: **7.25** m Max: **7.83** m
	Min: 2.12 m Avg: 2.25 m Max: 2.44 m	Min: **7.62** m Avg: **6.87** m Max: **8.33** m
	Min: 1.28 m Avg: 2.06 m Max: 2.44 m	Min: **6.55** m Avg: **7.51** m Max: **8.17** m
	Min: 0.00 m Avg: 0.63 m Max: 2.51 m	Min: **10.58** m Avg: **11.28** m Max: **14.96** m
	Min: 0.00 m Avg: 0.26 m Max: 1.29 m	Min: **6.91** m Avg: **12.55** m Max: **14.63** m
	Min: 0.00 m Avg: 0.67 m Max: 4.11 m	Min: **6.17** m Avg: **10.09** m Max: **13.25** m

IPA-II



IPA-III

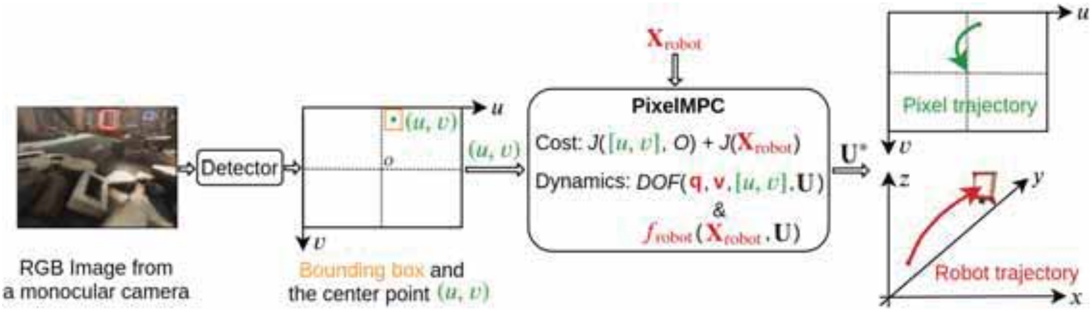


IPA-III

AI in Aerospace Systems

nature
At last — a computer program that can beat a champion Go player
ALL SYSTEMS GO
LOCKHEED MARTIN
ALPHA PILOT

IPA-IV: PixelMPC


RGB Image from a monocular camera
Detector
(u, v)
u
v
o
Bounding box and the center point (u, v)
X_robot
PixelMPC
Cost: J([u, v], O) + J(X_robot)
Dynamics: DOF(q, v, [u, v], U)
&
f_robot(X_robot, U)
U*
Pixel trajectory
Robot trajectory
z
y
x

Augmented Dynamics

Drone Dynamics

$$\dot{\mathbf{p}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = \mathbf{g} + m^{-1}(\mathbf{R}_b^{\omega}\mathbf{f}_T + \mathbf{f}_D + \mathbf{w_f})$$

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Pixel Dynamics

$$\begin{aligned} \dot{\mathbf{X}}_{\text{pixel}} &= F_{\text{pixel}}(\mathbf{q}, \mathbf{X}_{\text{pixel}}, \mathbf{U}) \\ &= \text{PolarToEuler}(DOF(\mathbf{q}, \mathbf{X}_{\text{pixel}}, \mathbf{U})) \end{aligned}$$

IPA-IV: PixelMPC

Outline

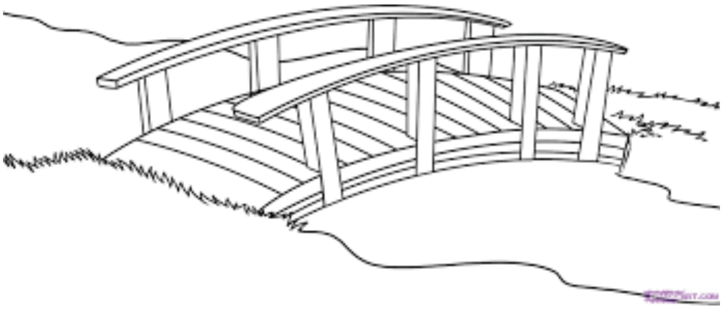
- Motivation & Intro
- Control Architectures & Uncertainty
- Control Architectures & Perception
- **Conclusions and Future**

Outline

- ✦ Motivation & Intro
- ✦ Control Architectures & Uncertainty
- ✦ Control Architectures & Perception
- ✦ Conclusions and Future

Decision Making Architectures

Partial Differential Equations



Stochastic Differential Equations

Deep Neural Network Architectures

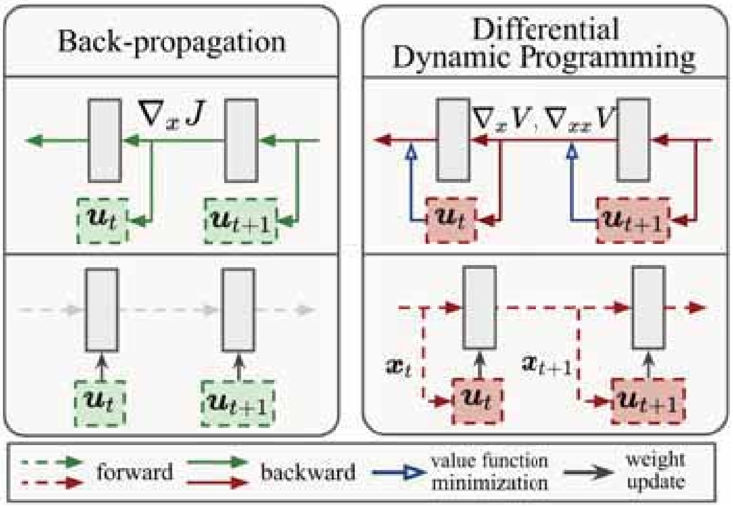
Stochastic Optimal Control ⟷ Forward/Backward Stochastic Differential Equations (FBSDEs)

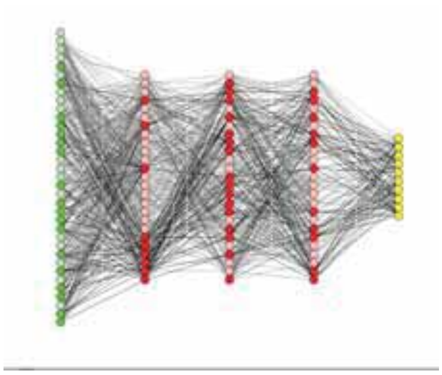
Perceptual Decision Making ⟷ Risk Measures and Stochastic Differential Games

Perceptual Decision Making ⟷ Control Barrier Functions & Barrier Certificates

Perceptual Decision Making ⟷ Adaptive Control & Contraction Theory

Safety & Deep Learning Theory


Back-propagation
Differential Dynamic Programming
$\nabla_x J$
u_t
u_{t+1}
$\nabla_x V, \nabla_{xx} V$
x_t
x_{t+1}
forward
backward
value function minimization
weight update



Optimal Control	Deep Learning
State	Output Activation
Controls	Weights
Time Horizon	Number of Layers
Terminal Cost	Loss Functions

Cost Function

$$\min_{\mathbf{u}} J(\bar{\mathbf{u}};\mathbf{x}_0) = \min_{\mathbf{u}} \left[\phi(\mathbf{x}_T) + \sum_{t=0}^{T-1} \ell_t(\mathbf{x}_t,\mathbf{u}_t)\right]$$

Dynamics

$$\mathbf{x}_{t+1} = f_t(\mathbf{x}_t,\mathbf{u}_t)$$

Autonomous Control and Decision Systems Lab

Students:

Collaborators:

Jim Regh - Georgia Tech

Naira Hovakimyan - UIUC

Ali-akbar Agha-mohammadi
JPL - NASA

NSF

Sandia National Laboratories

amazon web services

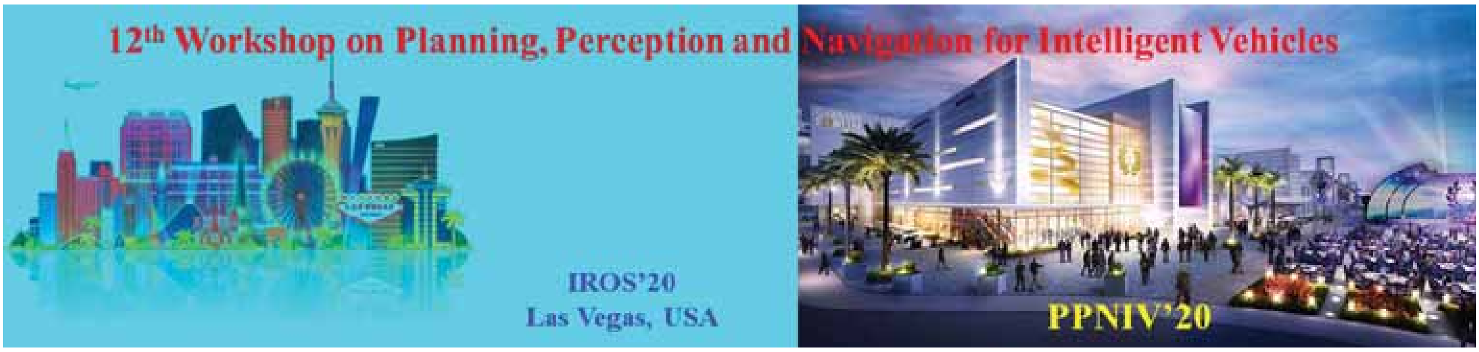
NASA

QUALCOMM

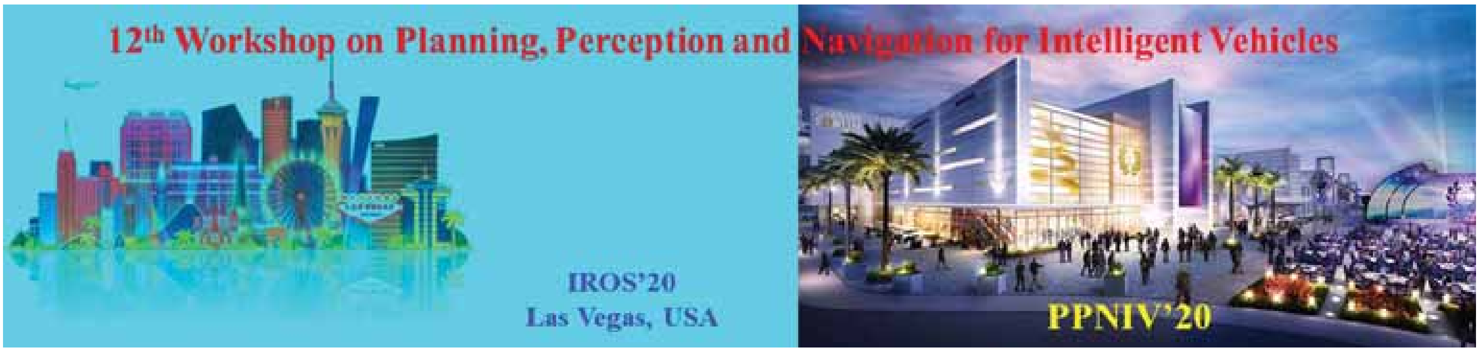
Vertical Lift Research
Center of Excellence

Georgia Tech Robotics & Intelligent Machines

KOMATSU


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Accepted papers

Title: Marker-Based Mapping and Localization for Autonomous Valet Parking
Authors: Zheng Fang, Yongnan Chen, Ming Zhou, Chao Lu

Title: Parameter Optimization for Loop Closure Detection in Closed Environments
Authors: Nils Rottmann, Ralf Bruder, Honghu Xue, Achim Schweikard, Elmar Rueckert

Title: Radar-Camera Sensor Fusion for Joint Object Detection and Distance Estimation in Autonomous Vehicles
Authors: Ramin Nabati, Hairong Qi

Title: SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving
Authors: Tiago Cortinhal, George Tzelepis, Eren Erdal Aksoy

Title: SDVTracker: Real-Time Multi-Sensor Association and Tracking for Self-Driving Vehicles
Authors: Shivam Gautam, Gregory P. Meyer, Carlos Vallespi-Gonzalez, Brian C. Becker

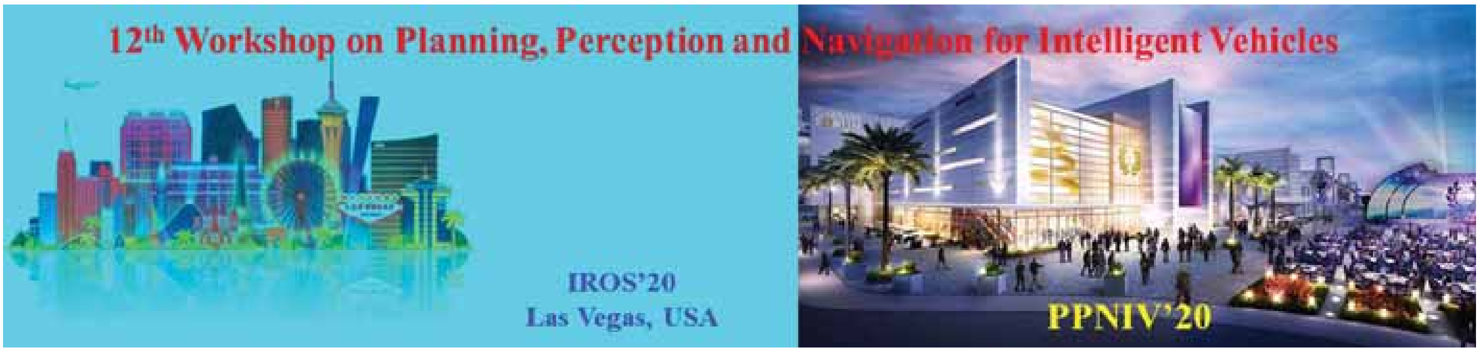
Title: Situation Awareness at Autonomous Vehicle Handover: Preliminary Results of a Quantitative Analysis
Authors: Tamas D. Nagy, Daniel A. Drexler, Nikita Ukhrenkov, Arpad Takacs, Tamas Haidegger

Title: Towards Context-Aware Navigation for Long-Term Autonomy in Agricultural Environments
Authors: M. Hollmann, B. Kisliuk, J.C. Krause, C. Tieben, A. Mocky, S. Putzy, F. Igelbrinky, T. Wiemanny, S. Focke Martinez, S. Stiene, J. Hertzberg

Title: Exploiting Continuity of Rewards – Efficient Sampling in POMDPs with Lipschitz Bandits
Authors: Ömer Sahin Tas, Felix Hauser, Martin Lauer

Title: Impact of Traffic Lights on Trajectory Forecasting of Human-driven Vehicles Near Signalized Intersections
Authors: Geunseob Oh, Huei Peng

Title: Semantic Grid Map based LiDAR Localization in Highly Dynamic Urban Scenarios
Authors: Chenxi Yang, Lei He, Hanyang Zhuang, Chunxiang Wang, Ming Yang


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

Marker-Based Mapping and Localization for Autonomous Valet Parking

Zheng Fang*, Yongnan Chen, Ming Zhou and Chao Lu

Abstract— **Autonomous valet parking (AVP) is one of the most important research topics of autonomous driving in low-speed scenes, with accurate mapping and localization being its key technologies. The traditional visual-based method, due to the change of illumination and appearance of the scene, easily causes localization failure in long-term applications. In order to solve this problem, we introduce visual fiducial markers as artificial landmarks for robust mapping and localization in parking lots. Firstly, the absolute scale information is acquired from fiducial markers, and a robust and accurate monocular mapping method is proposed by fusing wheel odometry. Secondly, on the basis of the map of fiducial markers that are sparsely placed in the parking lot, we propose a robust and efficient filtering-based localization method, which realizes accurate real-time localization of vehicles in parking lot. Compared with the traditional visual localization methods, we adopt artificial landmarks, which have strong stability and robustness to illumination and viewpoint changes. Meanwhile, because the fiducial markers can be selectively placed on the columns and walls of the parking lot, it is not easy to be occluded compared to the ground information, ensuring the reliability of the system. We have verified the effectiveness of our methods in real scenes. The experiment results show that the average localization error is about 0.3 m in a typical autonomous parking operation at a speed of 10km/h.**

I. INTRODUCTION

Autonomous valet parking is one of the most important research topics of autonomous driving in low-speed scenes. With the increasing density of vehicles in the city, parking space is tight and accidents are frequent during parking operations [1]. Autonomous valet parking technology can help realize high density parking, make full use of limited parking space, reduce accidents caused by human errors during parking, and also bring great convenience to drivers. After the vehicle is switched to AVP mode, it will automatically enter the parking lot to look for free parking spaces and park into any parking space available. However, the technology is quite far from mature yet and there are still many problems to be solved. One of the key problems is the lack of robust and accurate localization information in the absence of GNSS signals [2]. The schemes of traditional indoor localization technology , such as localization based on UWB, fixed laser scanners and other sensors [3], require a large number of high-cost modifications to the environment. Among SLAM-based self-localization methods, visual methods are preferred for its low cost compared to laser methods.

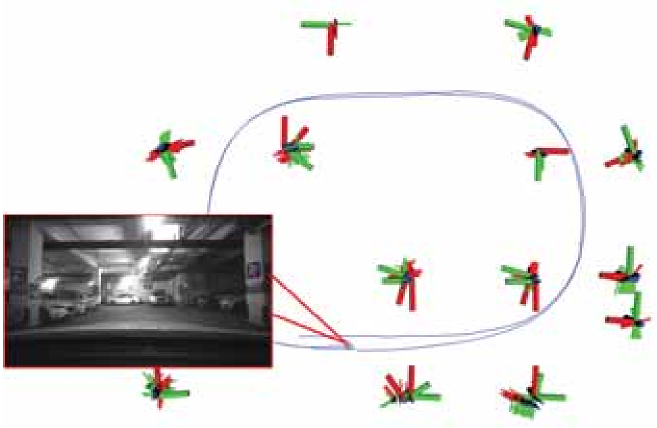


Fig. 1: The smaller figure in the bottom left corner shows a common scene in underground parking lot with complex illumination condition. Fiducial markers are applied to pillars and walls in this scene. As the bigger figure shows, our system realizes robust and accurate real-time localization in parking lot by fusing marker detections(as visualized in smaller figure) and odometry, with the help of a previously-built map of markers.

Visual SLAM can accurately estimate the current camera pose and establish the corresponding environmental map. ORB-SLAM2 [4], [5] and other feature-based methods have good results in the scene with rich texture. However, these methods suffer from environment appearances changes and complex illumination conditions. Thus, these methods could only provide visual maps that need to be established within a short period of time of localization usage, lacking long-term stability and practicability.

Fiducial marker [6] is a commonly used landmark, which is often used to estimate the pose of robots [7]. Compared with traditional geometric features, the fiducial marker has strong adaptability to illumination changes [8] and has larger identifiable angle range. In this paper, we propose a mapping and localization system based on fiducial markers, and utilize fiducial markers that are sparsely placed in real scenes and low-cost processors to realize accurate mapping and localization. Because the fiducial markers can be selectively placed on the columns and walls of the parking lot, it is not easy to be obscured compared to the ground information, which can ensure the reliability of the system. Also, due to the adoption of visual markers, this system only needs a low-performance ARM processor to realize robust localization,

This work is supported by the Major Science and Technology Projects Fundation of Liaoning Province (2019JH1/10100026), the Fundamental Research Funds for the Central Universities(No.N182608003, No.N172608005), Natural Science Foundation of Liaoning Province(20180520006)

Zheng Fang, Yongnan Chen, Ming Zhou and Chao Lu are with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China; Corresponding author: Zheng Fang, e-mail: fangzheng@mail.neu.edu.cn

which lays a foundation for the practicality of the system. The method proposed in this paper can establish a long-term stable and reusable parking lot map and provide accurate localization information for vehicles. We test the system on actual vehicles to verify the effectiveness and accuracy of our method. The experimental results show that the average localization error of our proposed methods is about 0.3m in the low-speed parking process with a vehicle speed of 10km/h. In summary, our main contributions are:

- Propose a robust and accurate marker-based mapping method by fusing scale information extracted from fiducial markers with odometry and feature points;
- Based on sparse fiducial marker map, propose a robust localization method with low computational resource consumption, by fusing marker detection and wheel odometry with a particle filter;
- Experiments in real scenes are carried out to verify the validity of our methods.

The rest of the paper is organized as follows. Section II describes the related works. Section III details the proposed mapping and localization methods. We validate our method in Section IV. Finally, Section V concludes the paper.

II. Related Works

In the past ten years, there have been many visual mapping and localization works in the field of AVP. According to the different emphases of these methods, we classify them into mapping and localization methods. Due to parking lots usually being private area, there are generally no maps established in advance, and vehicles need to establish their own maps. Visual SLAM-based mapping method is one of the commonly used methods. The V-Charge project [9] uses SFM framework to build a three-dimensional map of the environment through images collected by the multi-camera system configured by the vehicle. Chirca et al. [10] created a three-dimensional map of the environment through EKF-based visual SLAM. However, the above methods all utilize traditional geometric features, such as sparse points and straight lines in the environment. These traditional feature-based methods will be affected by changes in illumination, viewpoint and appearance when used in long term. In order to overcome these influences, high-dimensional environmental features are used for mapping. Huang et al [11] extract the ID information of the parking space through the fisheye camera, and established the semantic map of the parking lot environment by combining the monocular camera, wheel odometry and IMU. However, the parking space information in the parking lot is easily blocked by vehicles. Huang et al. additionally introduced visual tags to assist in localization. Similarly, Zong et al. [12] also introduce visual tags, combine with vehicle kinematics model, to improve the performance of ORB-SLAM in underground parking lots. In addition, road-based semantic features [13], [14], such as lane lines, speed bumps, turn signs and other features, are also applied to the mapping system. However, most of these ground semantic features may suffer from occlusion or be worn out in usage, which can lead to system failure. The computational consumption is also relatively high compared with traditional methods.

The vision-based localization methods [15]–[17] use the established map to obtain the pose of the camera relative to the map through descriptor matching. However, they are subject to localization failures in indoor parking lots and other low illumination environments [18]. Jeevan et al. [19] proposed a localization method, which fuse fiducial markers placed on the ground and wheel odometry. Compared with the feature-based method, it is more robust, but the map is generated by georeferencing each marker with GPS, thus the mapping can only be applied to outdoor scenes. For indoor parking lots, Qin et al. [20] utilize a variety of road semantic features and combined with wheel odometry to achieve centimeter-level parking accuracy. However, this method puts forward higher requirements for onboard hardware (high-performance processors, high-resolution cameras, etc.)

III. Approach

In this paper, we use monocular camera to get the image information. The monocular camera is installed to the center of the vehicle, behind the windshield to capture front-view scenes. Vehicle odometry information formed by steering wheel angle and vehicle speed is also used in our system. Intrinsic and extrinsic parameters of all sensors were calibrated offline in advance.

The framework consists of two parts, as shown in Fig.2. The first part is mapping, in which we use the front-view monocular camera to detect fiducial markers, extract scale and pose information and then fuse with odometry data to build a global fiducial marker map. This marker map is saved for localization. Then the vehicle is localized by matching fiducial markers extracted from monocular image to the marker map. In the end, a particle filter fuses visual localization results with odometry, which guarantees the system survives in the marker-less region and has a smooth output.

A. Mapping with Fiducial Markers and Vehicle Odometry

The proposed mapping method contains three main modules: tracking, local mapping and loop closing [21]. In the initialization part of the tracking module, we use fiducial markers to recover the scale of monocular camera. In the local mapping module, the map is extended by adding newly observed markers and new map points. In addition, the poses of local keyframes and local map points are optimized jointly in this module, called local bundle adjustment(BA). Accumulated drift will be eliminated by loop closing.

1) Scale Recovery From Visual Fiducial Markers: There are many different kinds of visual fiducial markers. We choose ArUco marker in our system due to its robustness and high-efficiency and it is included in OpenCV [22].

Adding the ArUco marker to monocular SLAM [23], [24] solves the problem of scale ambiguity. At initialization, we can recover the scale factor s of the monocular camera trajectory by obtaining the same ArUco Marker observed

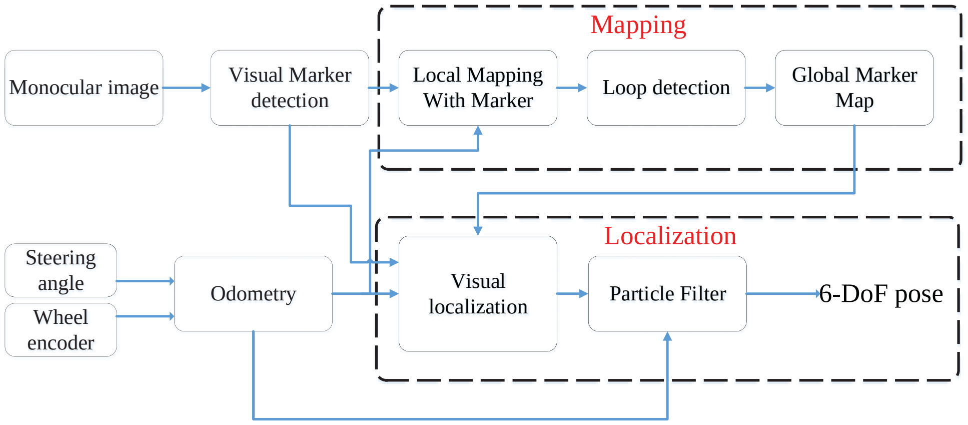


Fig. 2: Block diagram illustrating the full pipeline of the proposed system. In the mapping procedure, it builds a map of the large-scale indoor parking lot with fiducial markers. Based on this prior map, the localization procedure can provide precise 6-DoF pose through a particle filter fusing fiducial markers with odometry.

by both keyframes as follows.

$$R^{k+1}_{WC}R^{k+1}_{CM_i}t^{k+1}_{M_iC} - R^{k}_{WC}R^{k}_{CM_i}t^{k}_{M_iC} = s(t^{k+1}_{WC} - t^{k}_{WC}) \tag{1}$$

where $[R^k_{CM_i}\ t^k_{CM_i}]$ is the pose from ith marker to camera at frame k and $[R^k_{WC}\ t^k_{WC}]$ is the pose from camera to world at frame k.

2) Pose Optimization with Vehicle Odometry Constraints: Most cars are equipped with wheel encoders [25]. In most cases, wheel encoders provide a reliable measurement of the distance traveled by the wheel. In our case, we can directly read the wheel speed v of the rear wheels and the corresponding steering wheel angle δ through the vehicle's CAN bus. Then we can get the pose R^{k+1}_{WV} and t^{k+1}_{WV} of the vehicle for frame $k+1$ according to the vehicle odometry as Equation 2.

$$\begin{cases} x_{k+1} = x_k + \Delta x cos(\theta_k) - \Delta y sin(\theta_k) \\ y_{k+1} = y_k + \Delta x sin(\theta_k) + \Delta y cos(\theta_k) \\ \theta_{k+1} = \theta_k + \Delta\theta \end{cases} \tag{2}$$

Therefore, in local map optimization, we additionally introduce vehicle odometry error term on top of the reprojection error term by Equation 3

$$\begin{aligned} &\gamma = \{R^j_{CW}, t^j_{CW}\} \\ &\gamma^* = argmin(\textstyle\sum_k E_{proj}(k,j) + E_{vehicle}(i,j)) \end{aligned} \tag{3}$$

where E_{proj} is the reprojection error of current frame j for given match k . And the vehicle odometry error term $E_{vehicle}$ between keyframe i and j is denoted by Equation 4.

$$\begin{aligned} &E_{vehicle}(i,j) = \rho([e_R^T e_t^T]\textstyle\sum_I [e_R^T e_t^T]^T) \\ &e_R = Log((Exp(w_v \Delta t(i,j)))^T R^i_{VW} R^j_{WV}) \\ &e_p = R^i_{VW}(t^j_{WV} - t^i_{WV}) - v^i_{WV}\Delta t \end{aligned} \tag{4}$$

where $\rho(\cdot)$ is the robust Huber cost function, $\sum_I$ is the information matrix of vehicle odometry error term.

B. Marker Map-based Real-time Localization

To better suit the need of performing real-time localization on automotive-grade embedded processors, we propose a particle filter-based method to fuse visual and odometry information for localization in indoor parking zone. The marker map we use is created in previous part.

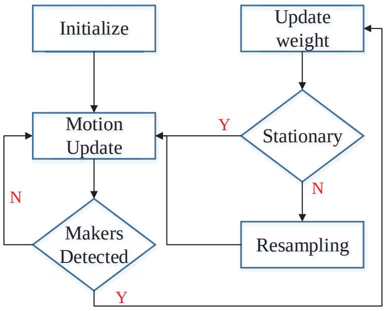


Fig. 3: Localization Algorithm Structure

1) Initialization: Our system requires at least one marker detection to initialize. After localization system is started, markers in surrounding area are detected by vision and their IDs and relative poses to vehicle are used for initialization. With known ID, a marker's absolute pose can be acquired from marker map created earlier by matching its ID to the marker with same ID in the map. Then the vehicle's initial pose in map coordinate can be calculated from marker's absolute pose and marker's relative pose to vehicle. Being vehicle pose in 2D space (x, y, θ), for kth marker detected during initialization with a relative pose to vehicle as (x', y', θ'), its absolute pose in map coordinate being $m_k = [x_k\ y_k\ \theta_k]^T$, then

$$\begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} x_k - x' cos\theta' + y' sin\theta' \\ y_k - x' sin\theta' + y' cos\theta' \\ \theta_k - \theta' \end{bmatrix} \tag{5}$$

where $X_0 = [x_0\ y_0\ \theta_0]^T$ is vehicle's initial pose under map coordinate.

To improve initialization accuracy, we do this calculation multiple times and use the results' average as vehicle initial pose, and then generate our set of particles with different poses (x_n, y_n, θ_n) around this pose according to normal distribution, where $(std_x, std_y, std_\theta)$ are preset initialization parameters.

2) Motion Update: By using wheel odometry we can obtain vehicle's relative movements sequentially and use them to perform a prediction update to our particles. Because in our system the frequency of marker observations is significantly lower than that of odometry feedbacks, so the motion and observation could be considered separately while updating state probability.

If at moment t-1 the nth particle's pose state is $X_{t-1}^n = [x_{t-1}^n \ y_{t-1}^n \ \theta_{t-1}^n]^T$, then after a motion update its state is

$$\begin{gathered}\begin{bmatrix} x_t^n \\ y_t^n \\ \theta_t^n \end{bmatrix} = \begin{bmatrix} \mathrm{N}\left(0, x_\sigma^2\right) \\ \mathrm{N}\left(0, y_\sigma^2\right) \\ \mathrm{N}\left(0, \theta_\sigma^2\right) \end{bmatrix} + \\ \begin{bmatrix} x_{t-1}^n + \cos\left(\theta_{t-1}^n + \Delta\theta\right)\Delta x - \sin\left(\theta_{t-1}^n + \Delta\theta\right)\Delta y \\ y_{t-1}^n + \sin\left(\theta_{t-1}^n + \Delta\theta\right)\Delta x \\ \theta_{t-1}^n + \Delta\theta \end{bmatrix}\end{gathered} \tag{6}$$

where $u_t = [\Delta x \ \Delta y \ \Delta\theta]^T$ are translation and rotation increments measured by wheel odometry, and $(x_\sigma, y_\sigma, \theta_\sigma)$ are preset motion noises. Motion noises is estimated by experiments and in our case set to $(0.005m, 0.005m, 0.001rad)$.

3) Observation Update: While doing motion updates the marker detector is detecting markers in monocular images at the same time. Due to the pose ambiguity problem, marker detection's accuracy is slightly lower on longer distance, so we only take detections within a distance threshold($10m$). Once one or more such detections are returned, observation update is carried out using these detections.

To evaluate the error of each particle's pose to vehicle's actual pose, based on relative pose of marker to vehicle measured by marker detector and each particle's pose state, we calculate the marker poses observed from each particle and compare them to this marker's actual pose in the map. If the relative pose of a marker to vehicle is measured as $Z_t = [x_{ob} \ y_{ob} \ \theta_{ob}]^T$, then the nth particle in particle set, with a pose state of $X_n = [x_n \ y_n \ \theta_n]^T$, the marker pose observed from this particle under map coordinate is:

$$\begin{bmatrix} x_{ob\_\mathrm{map}} \\ y_{ob\_\mathrm{map}} \\ \theta_{ob\_\mathrm{map}} \end{bmatrix} = \begin{bmatrix} x_{ob}\cos\theta_n - y_{ob}\sin\theta_n + x_n \\ x_{ob}\sin\theta_n + y_{ob}\cos\theta_n + y_n \\ \theta_{ob} + \theta_n \end{bmatrix} \tag{7}$$

The particle's weight can be calculated using error between this pose and the marker's real pose, thus the weight w_n of nth particle is:

$$w_n = \frac{e^{-\left(\left(\left(x_{ob\_map} - x_{ob}\right)^2/\sigma_x^2\right) + \left(\left(y_{ob\_map} - y_{ob}\right)^2/\sigma_y^2\right)\right)/2}}{2\pi\sigma_x\sigma_y} \tag{8}$$

where σ_x and σ_y are observation noises, set to 0.3m and 0.3m in our case respectively. They are set slightly bigger than translational errors of marker detections intendedly. Considered that markers in parking garage is relatively sparse, this can help the system correct odometry accumulation errors more gradually and avoid local sharp changes in pose output, which may have negative effects on motion control and path planning. Then we complete weights update by normalize weights of the whole particle set:

$$w_{\mathrm{norm}} = \frac{w_{1:n}}{\sum_{n=1}^{n=num} w_n} \tag{9}$$

Where num is particles' number and w_{norm} is the array of normalized particle weights.

4) Resampling: With particle weights updated, we firstly check vehicle's moving state through latest odometry readout. If odometry shows that vehicle is stationary, we keep the particle weights update without resampling because vehicle's pose is not supposed to change at this moment. If the vehicle is moving, we resample the particles by their weights and reinitialize weights of the new particle set as:

$$w_{1:num} = \frac{1}{num} \tag{10}$$

The X_t probability distribution is approximated by the new particle set.

Finally, we output vehicle pose every time system state is updated. To smooth estimated trajectory and reduce pose jumps, we choose average pose of all particles $(x_{avg}, y_{avg}, \theta_{avg})$ as the final pose output.

IV. Experiments and Analysis

In order to verify the validity of the system proposed in this paper, we conducted separate experiments for mapping and real-time localization. The experiment environment is an underground parking garage with uneven lighting and an area of about 500 m^2. Twenty markers of 0.552m*0.552m size are placed on the walls and pillars of parking lots where they are easy to be observed and not easily to be obscured by the vehicles in the parking lot. The average interval between the markers is about 8m, excluding the case where there are multiple markers on different sides of the same pillar. The vehicle is equipped with two wheel encoders, an Intel RealSense D435i camera , VLP16 LiDAR and an embedded platform with Ubuntu 16.04. The vehicle travels at a constant speed of 10km/h. The video link for a demonstration of the proposed system is: `https://youtu.be/1lr3eRAjFVA`

A. Mapping Metric Evaluation

For mapping metric evaluation, considering the high accuracy and robustness and maturity of the 3D laser SLAM algorithm in indoor scenarios, we collect laser point cloud data during the experiment. We use the Lego-Loam [26] algorithm to process the acquired data and treat the resulting laser trajectory as ground truth. The total trajectory length is 143 m. We recorded the camera trajectory as well as the laser trajectory. Due to the uniqueness of our sensor configuration, it is hard to directly compare against other existing algorithms. We compared our method with ground truth in terms of mapping accuracy.

The mapping result and estimated trajectories are shown in Fig. 4. The RMSE of absolute trajectory error is 0.438m and

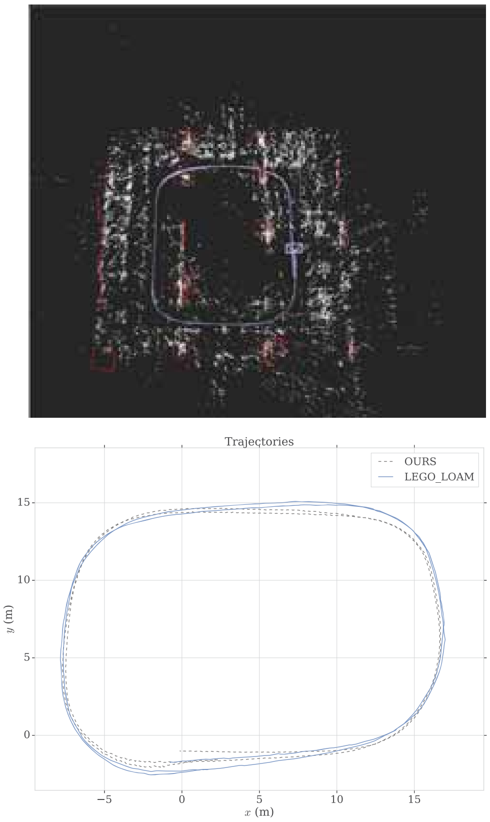


Fig. 4: (top) Map of the parking lot. The red squares are visual fiducial markers. (bottom) Estimated trajectories by our methods and the ground truth.

the normalized estimation error squared(NESS) is 0.306%. It can be seen that our algorithm performs well by fusing feature points, visual fiducial markers and vehicle odometry.

B. Localization Accuracy Evaluation

After the marker map is created, we performed real-time localization experiments in underground parking garage mentioned above, as shown in Fig. 1, where the blue line is estimated trajectory, the bigger axes show the absolute poses of markers in the map and smaller axes show the poses of markers observed from different particles. We also use trajectories estimated by laser SLAM as ground truth to evaluate accuracy of our localization method. The results of two independent experiments are shown in Fig. 5, Fig. 6 and TABLE I.

TABLE I: Errors in two experiments

Error	Mean[m]	Max[m]	Min[m]	RMSE[m]
Experiment 1	0.301	0.775	0.0153	0.347
Experiment 2	0.264	0.687	0.0248	0.307

As shown above, our method is low on most of the errors, with an average error around 0.3m. The localization performance is also stable throughout the whole trajectory. Due to the fact that marker landmarks are sparsely distributed in the parking garage(as stated above, the average distance between markers is 8m), localization error at some places with few or no markers detectable will be slightly bigger, especially during turns(as shown at the upper right and bottom left corners of trajectories in Fig. 5), but these errors are still acceptable and could be corrected quickly as the trajectories show.

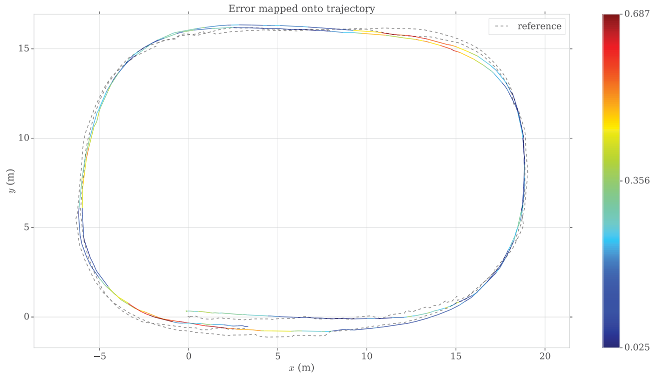


Fig. 5: Comparison of marker-based localization and ground truth(grey line)

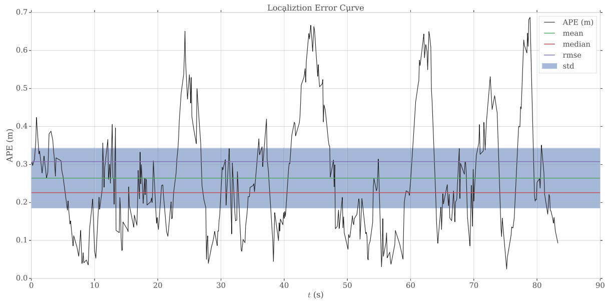


Fig. 6: Localization error graph of two experiments respectively

TABLE II: Running performance of our method on different hardware

	CPU occupation	Memory used[MB]	Frequency[Hz]
i7 laptop	14%	500	100
A53 embedded	25%	510	100

C. Computational Resource Demand

As mentioned above, our localization method is developed for online usage on intelligent vehicle's onboard embedded processor, so the algorithm's computational resource demand needs to be as low as possible. We tested our algorithm on 8-core i7-7700HQ equipped laptop and 4-core A53 equipped embedded platform respectively. As shown in TABLE II, while performing localization successfully, the computational resource consumption of our method is also suitable for real-time application on intelligent vehicles.

D. System Robustness

To verify our method's environment robustness over feature point-based methods, we tested these methods in long-term localization. Experiments showed that after significant changes occurred in operation environment, appearance changes in certain locations will lead to false feature point matches to map(as shown in Fig. 7), causing localization failures. On contrary, because marker detections are not affected by appearance changes of surrounding area(as shown in Fig. 8), localization based on marker map is still robust and effective.

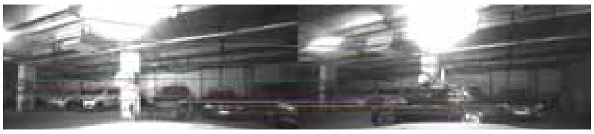


Fig. 7: The false feature point matches of the same place at different time due to appearance changes

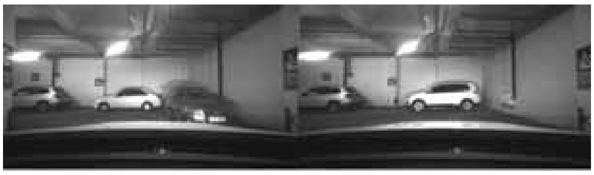


Fig. 8: Marker detections are not affected by changes in surrounding area

V. CONCLUSIONS

In order to realize the robust localization for autonomous parking in underground parking lots, we introduce visual fiducial marker as a stable artificial landmark to establish a robust and long-term usable map. On this basis, an efficient localization algorithm based on particle filter is proposed to perform robust and accurate localization. However, The method we proposed still requires manually placing markers in the parking lot. In the future, we plan to replace fiducial markers with the existing text landmarks in the parking lot to further improve the practicability of our system.

REFERENCES

[1] H. Banzhaf, D. Nienhuser, S. Knoop, and J. Marius Zollner, "The future of parking: A survey on automated valet parking with an outlook on high density parking," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 1827–1834, 2017.

[2] B. Li, L. Yang, J. Xiao, R. Valde, M. Wrenn, and J. Leflar, "Collaborative Mapping and Autonomous Parking for Multi-Story Parking Garage," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1629–1639, 2018.

[3] H. M. Hussien, Y. N. Shiferaw, and N. B. Teshale, "Survey on indoor positioning techniques and systems," in *Information and Communication Technology for Development for Africa*, pp. 46–55, Springer, 2018.

[4] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[5] J. B. M. L. Felix Nobis, Odysseas Papanikolaou, "Persistent map saving for visual localization for autonomous vehicles: An orb-slam extension," in *International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2020.

[6] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, pp. 38–47, 2018.

[7] H. Lim and Y. S. Lee, "Real-time single camera slam using fiducial markers," in *2009 ICCAS-SICE*, pp. 177–182, IEEE, 2009.

[8] D. Hu, D. Detone, and T. Malisiewicz, "Deep charuco: Dark charuco marker pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8436–8444, 2019.

[9] U. Schwesinger, M. Burki, J. Timpner, S. Rottmann, L. Wolf, L. M. Paz, H. Grimmett, I. Posner, P. Newman, C. Hane, L. Heng, G. H. Lee, T. Sattler, M. Pollefeys, M. Allodi, F. Valenti, K. Mimura, B. Goebelsmann, W. Derendarz, P. Muhlfellner, S. Wonneberger, R. Waldmann, S. Grysczyk, C. Last, S. Bruning, S. Horstmann, M. Bartholomaus, C. Brummer, M. Stellmacher, F. Pucks, M. Nicklas, and R. Siegwart, "Automated valet parking and charging for e-mobility," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 157–164, 2016.

[10] M. Chirca, R. Chapuis, and R. Lenain, "Autonomous Valet Parking System Architecture," in *Proceedings of IEEE Conference on Intelligent Transportation Systems, ITSC*, pp. 2619–2624, 2015.

[11] Y. Huang, J. Zhao, X. He, S. Zhang, and T. Feng, "Vision-based Semantic Mapping and Localization for Autonomous Indoor Parking," in *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 636–641, 2018.

[12] W. Zong, L. Chen, C. Zhang, Z. Wang, and Q. Cheny, "Vehicle model based visual-tag monocular ORB-SLAM," in *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, pp. 1441–1446, 2017.

[13] A. Ranganathan, D. Ilstrup, and T. Wu, "Light-weight localization for vehicles using road markings," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 921–927, 2013.

[14] Y. Lu, J. Huang, Y. T. Chen, and B. Heisele, "Monocular localization in urban environments using road markings," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 468–474, 2017.

[15] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale, "Summary Maps for Lifelong Visual Localization," *Journal of Field Robotics*, vol. 33, no. 5, pp. 561–590, 2016.

[16] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 719–724, 2013.

[17] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, M. Haueis, and C. Stiller, "Video based localization for Bertha," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 1231–1238, 2014.

[18] P. Nelson, W. Churchill, I. Posner, and P. Newman, "From dusk till dawn: Localisation at night using artificial light sources," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 5245–5252, 2015.

[19] P. Jeevan, F. Harchut, B. Mueller-Bessler, and B. Huhnke, "Realizing autonomous valet parking with automotive grade sensors," in *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3824–3829, 2010.

[20] T. Qin, T. Chen, Y. Chen, and Q. Su, "AVP-SLAM: Semantic Visual Mapping and Localization for Autonomous Vehicles in the Parking Lot," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, accepted.

[21] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A versatile visual SLAM framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2292–2295, 2019.

[22] G. Bradski, "The OpenCV Library," *Dr Dobbs Journal of Software Tools*, vol. 25, 2000.

[23] R. Munozsalinas, M. J. Marinjimenez, E. Yeguasbolivar, and R. Medinacarnicer, "Mapping and localization from planar markers," *Pattern Recognition*, vol. 73, pp. 158–171, 2018.

[24] R. Muñoz-Salinas and R. Medina-Carnicer, "Ucoslam: Simultaneous localization and mapping by fusion of keypoints and squared planar markers," *Pattern Recognition*, vol. 101, p. 107193, 2020.

[25] H. Banzhaf, D. Nienhuser, S. Knoop, and J. Marius Zollner, "The future of parking: A survey on automated valet parking with an outlook on high density parking," pp. 1827–1834, 2017.

[26] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758–4765, IEEE, 2018.

Parameter Optimization for Loop Closure Detection in Closed Environments

Nils Rottmann[1], Ralf Bruder[1], Honghu Xue[1], Achim Schweikard[1], Elmar Rueckert[1]

Abstract— **Tuning parameters is crucial for the performance of localization and mapping algorithms. In general, the tuning of the parameters requires expert knowledge and is sensitive to information about the structure of the environment. In order to design truly autonomous systems the robot has to learn the parameters automatically. Therefore, we propose a parameter optimization approach for loop closure detection in closed environments which requires neither any prior information, e.g. robot model parameters, nor expert knowledge. It relies on several path traversals along the boundary line of the closed environment. We demonstrate the performance of our method in challenging real world scenarios with limited sensing capabilities. These scenarios are exemplary for a wide range of practical applications including lawn mowers and household robots.**

I. Introduction

Algorithms for simultaneous localization and mapping (SLAM) [9], [2] such as FastSLAM [24], GMapping [13], [12] or RTabMap [17], [18] require the tuning of a large number of parameters. A correct setting of these parameters is crucial for the performance of these algorithms [1]. In general, finding convenient parameters for a certain mapping task requires prior knowledge on the structure of the environment and the robot itself. However, truly autonomous systems are expected to be able to adapt themselves to any environment and thus, being able to learn the required parameters autonomously. A well-known method for such meta-parameter learning problems is classical Reinforcement Learning (RL) [31], more specifically Bayesian Optimization (BO) [29], [30]. BO is a black box optimizer that only requires a definition of a cost function. A proper definition of the cost function is critical for the success of the parameter learning procedure. For mapping algorithms, a natural choice would be to define the cost as the difference between the estimated map and the respective ground truth. However, the ground truth is not known a priori such that other cost measures have to be developed for the meta-parameter learning.

An area of increasing importance in the last decade is the field of low-cost robotics [7], [15]. Robots such as lawn mowers or vacuum cleaners are used ubiquitously in households and work exclusively in closed environments, e.g. on a lawn or in an apartment. In general, these robots have only limited sensing capabilities due to the low-cost design. Algorithms dealing with the mapping problem for this type of robots are proposed in [26] and [6], where sonars or infrared sensors are used and linear features required. An indoor mapping approach using a wall following scheme has been presented in [34], where map rectification has been used under the assumption of straight wall segments.

Where there is an active research for SLAM approaches for autonomous vacuum cleaner, e.g. vision SLAM [16], [19], [20], autonomous lawn mowers still move randomly within the area of operation. Thereby, they use a boundary wire enclosing the working area which emits an electromagnetic signal that can be detected by the robot. Towards efficient localization and planning, a first step can be taken by mapping the enclosure. In [10], a map generation approach based the loop closure detected by returning to the home station has been introduced. Thereby, the lawn mower was driving along the boundary wire while measuring movements with the wheel odometry. However, using only a single loop closure requires to distribute the error along all estimated positions equally. Hence, detecting additional loop closures is favorable for a robust mapping approach. In [28], the authors proposed a loop closure detection approach for low-cost robots based on odometry data only. The data is collected when the robot is following the boundary of the closed environment. The performance of this approach depends highly on the correct meta-parameter setting which requires a priori knowledge about the closed environment. Hence, to enable truly autonomous behavior the robot has to learn the parameter by itself such that it can adapt to any arbitrary closed environment. Therefore, we developed a RL approach for learning meta-parameters under the assumption that the average distance traveled by the robot along a closed environment is equal to its circumference. We demonstrate the performance and robustness of our approach in different challenging simulation and real world scenarios.

The contributions of the paper are three-fold. First, we adapt and improve the method introduced in [28] by introducing relative error measurements for each loop closure using the *Iterative Closest Point* (ICP) approach [4]. Second, we insert a feasibility check in order to cope with recurrent symmetric structures and third, we introduce a RL scheme for learning the meta-parameters to enable true autonomous behavior. For our approach, we require that the robot is able to travel several times along the boundary line of the closed environment, e.g. by using a perimeter wire.

We start by summarizing and adapting the mapping method

[1] Institute for Robotics and Cognitive Systems, University of Luebeck, Ratzeburger Allee 160, 23562 Luebeck, Germany {rottmann, bruder, schweikard, rueckert}@rob.uni-luebeck.de

from [28], Section II. In Section III, we derive the RL procedure for meta-parameter learning. The procedure is divided into two stages, parameter learning for loop closure detection and pose graph optimization. We evaluate our approach in simulations and on real data in Section IV and in Section V we conclude.

II. Mapping Procedure

As the robot follows the boundary line of the closed environment, e.g. by means of the electromagnetic wire signal, a path based on the robot odometry data can be recorded. This path can then be transferred into the well-known pose graph representation [11]. Loop closures can be identified by comparing the neighborhoods of the pose graph vertices with each other, e.g. due to shape comparison. Based on the identified loop closing constraints the pose graph can be optimized by reducing the sum of weighted residual errors. Both, finding good loop closing constraints and optimizing the pose graph are strongly dependent on the correct parameter tuning. In the following, we shortly recapture pose graph representation as well as the general idea for detecting loop closing constraints. In Table I, we listed our notations for the different variables used throughout this paper.

A. Pose Graph Representation

Let $\boldsymbol{p} = \{\boldsymbol{p}_0, \dots, \boldsymbol{p}_N\}$ be a set of $N+1$ poses representing the position and orientation of a mobile robot in a two dimensional space, hence $\boldsymbol{p}_i = [\boldsymbol{x}_i^\top, \varphi_i]^\top$. Here, $\boldsymbol{x}_i \in \mathbb{R}^2$ is the cartesian position of the robot and $\varphi_i \in [-\pi, \pi]$ the corresponding orientation as an euler angle with the integer $i = 0{:}N$. The relative measurement between two poses i and j is then given as

$$\boldsymbol{\xi}_{ij} = \begin{bmatrix} \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i) \\ \varphi_j - \varphi_i \end{bmatrix} = \boldsymbol{p}_j \ominus \boldsymbol{p}_i, \tag{1}$$

where $\boldsymbol{R}_i = \boldsymbol{R}_i(\varphi_i)$ is a planar rotation matrix and $\ominus$ the pose compounding operator introduced by [21]. The pose graph is then a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $N+1$ vertices, representing the poses, and $N+M$ edges, representing the relative pose measurements. In our case, these pose measurements are composed of N odometric constraints and M loop closing constraints. In Figure 1, an example of a pose graph with four odometric and one loop closing constraint is shown. The connection between the vertices by the edges can be compactly written using an incident matrix $\boldsymbol{A}$, which is exemplarily shown on the right in Figure 1.

TABLE I: Variable definitions used throughout this paper.

$\boldsymbol{p}$	$\mathbb{R}^3$	poses
$\boldsymbol{x}$	$\mathbb{R}^2$	positions in meters
φ	$\mathbb{R}$	orientations in rad
$\boldsymbol{R}$	$\mathbb{R}^{2\times2}$	two-dimensional rotation matrix
$\boldsymbol{\xi}$	$\mathbb{R}^3$	relative measurements
$\boldsymbol{P}$	$\mathbb{R}^{3\times3}$	cov. matrix to the noise of the rel. measurements
N	$\mathbb{N}$	number of odometric constraints
M	$\mathbb{N}$	number of loop closing constraints
L_{NH}	$\mathbb{R}$	neighborhood length in meters
c_{min}	$\mathbb{R}$	minimum comparison error
γ_1, γ_2	$\mathbb{R}$	pose graph optimization parameters
U	$\mathbb{R}$	circumference of the closed environment in meters
u	$\mathbb{R}$	path distance between loop closing pairs in meters
$\Delta\varphi$	$\mathbb{R}$	difference in orientation in rad
φ_{cycle}	$\mathbb{R}$	feasibility check parameter

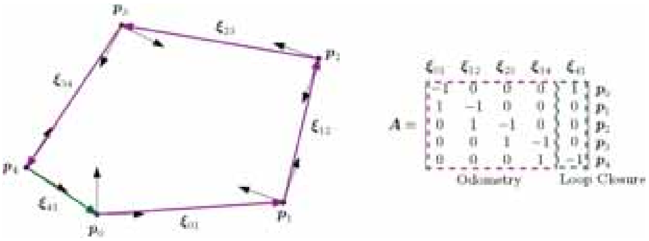

p_3
ξ_{23}
p_2
ξ_{34}
ξ_{12}
p_4
ξ_{41}
p_0
ξ_{01}
p_1
$\boldsymbol{A} =$
ξ_{01} ξ_{12} ξ_{23} ξ_{34} ξ_{41}
-1 0 0 0 1 p_0
1 -1 0 0 0 p_1
0 1 -1 0 0 p_2
0 0 1 -1 0 p_3
0 0 0 1 -1 p_4
Odometry
Loop Closure

Fig. 1: Pose graph with five vertices connected with five edges. Four of the edges are odometric constraints and one is a loop closing constraint. On the right, the incidence matrix is shown divided into the parts containing the odometric constraints and the loop closing constraints.

To account for noise in the relative pose measurements, we include zero mean Gaussian noise $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}_{ij})$, where

$$\hat{\boldsymbol{\xi}}_{ij} = \boldsymbol{\xi}_{ij} + \boldsymbol{\epsilon}_{ij}, \tag{2}$$

denotes the with noise corrupted relative pose measurements. The overall optimization problem is then to minimize the sum of weighted residual errors $\boldsymbol{r}_{ij}(\boldsymbol{p})$ with respect to the pose estimates $\boldsymbol{p}$,

$$\min_{\boldsymbol{p}} \sum_{(i,j)\in\mathcal{E}} ||\boldsymbol{r}_{ij}(\boldsymbol{p})||^2_{\boldsymbol{P}_{ij}}, \tag{3}$$

where

$$||\boldsymbol{r}_{ij}(\boldsymbol{p})||^2_{\boldsymbol{P}_{ij}} = [(\boldsymbol{p}_j \ominus \boldsymbol{p}_i) - \hat{\boldsymbol{\xi}}_{ij}]^\top \boldsymbol{P}_{ij}^{-1} [(\boldsymbol{p}_j \ominus \boldsymbol{p}_i) - \hat{\boldsymbol{\xi}}_{ij}]. \tag{4}$$

Here, $\boldsymbol{P}_{ij}$ is the covariance matrix corresponding to the noise of the relative measurements $\hat{\boldsymbol{\xi}}_{ij}$.

B. Loop Closure Detection

Based on the pose graph, loop closing constraints are detected by comparing the shape of the neighborhood regions of each vertex with another. Therefore, a piecewise linear function

$$\theta(x) = \phi_i \quad \text{for} \quad l_{i-1} \le x < l_i, \quad i = 0, 1, \dots, N. \tag{5}$$

representing the shape of the pose graph is constructed by accumulating the orientation and distance differences between the poses

$$\begin{aligned} \phi_i &= \phi_{i-1} + \Delta\phi_i \\ l_i &= l_{i-1} + ||\boldsymbol{v}_i||. \end{aligned} \tag{6}$$

Here, $\boldsymbol{v}_i = \boldsymbol{x}_i - \boldsymbol{x}_{i-1}$ and $\Delta\phi_i = \varphi_i - \varphi_{i-1}$ starting by $\phi_0 = \varphi_0$ and $l_0 = 0$. Figure 2 shows such a constructed piecewise orientation function. By defining the neighborhood of a vertex i as $[l_i - L_{\text{NH}}, l_i + L_{\text{NH}}]$, a comparison error between two vertices i and j is given as

$$\boldsymbol{C}_{ij} = \int_{-L_{\text{NH}}}^{+L_{\text{NH}}} [\theta(l_i + x) - \phi_i] - [\theta(l_j + x) - \phi_j] \, \mathrm{d}x. \tag{7}$$

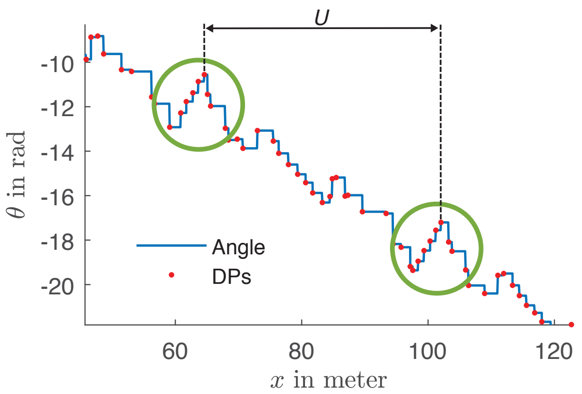


Fig. 2: Example for the piecewise linear orientation function $\theta(x)$. The green circled regions show similar path segments. The vertices or dominant points (DPs) of the pose graph are pictured as red dots. The estimated circumference U for the closed environment is exemplarily depicted for a possible loop closing pair.

We rewrite Equation (7) as a sum over m linearly distributed evaluation points

$$\boldsymbol{C}_{ij} = \frac{1}{m}\sum_{k=1}^{m}\left[\theta(l_i + x_k) - \phi_i\right] - \left[\theta(l_j + x_k) - \phi_j\right] \tag{8}$$

with $x_1 = -L_{\text{NH}}$, $x_m = +L_{\text{NH}}$. In Figure 3, a resulting error matrix between all vertices is graphically illustrated. A loop closing pair $SP_k = \{\boldsymbol{p}_i, \boldsymbol{p}_j\}$ for $i \neq j$ is defined as a local minimum of C_{ij} for which holds $C_{ij,\text{min}} < c_{\text{min}}$. A local minimum represents thereby the best possible loop closure in a certain region of the error matrix and the threshold c_{min} ensures that not every local minimum is selected as loop closing pair, but only sufficient accurate ones. Thus, the parameters L_{NH} and c_{min} are crucial for efficiently finding convenient loop closing pairs and will be learned through Bayesian Optimization. This process is discussed in Section III-A.

After detecting a loop closure between the vertices i and j of the pose graph, the loop closing constraint as a relative measurement $\hat{\xi}_{ij}$ has to be added. Therefore, the neighborhood regions of both poses i and j are discretized as distinct points, represented by the sets $X_i = \{\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,K}\}$ and $X_j = \{\boldsymbol{x}_{j,1}, \ldots, \boldsymbol{x}_{j,K}\}$, and transformed such that both poses i and j are equal with $\hat{\boldsymbol{p}}_i = \hat{\boldsymbol{p}}_j = [0, 0, 0]^\top$. By using an adapted ICP approach [3], which minimizes the distance error

$$\min_{\boldsymbol{R}_\beta, \boldsymbol{t}} E_{\text{dist}}(\boldsymbol{R}_\beta, \boldsymbol{t}) = \min_{\boldsymbol{R}_\beta, \boldsymbol{t}} \sum_{k=1}^{K} ||\boldsymbol{R}_\beta \boldsymbol{x}_{i,k} + \boldsymbol{t} - \boldsymbol{x}^*_{i,k}|| \tag{9}$$

with $\boldsymbol{x}^*_{i,k}$ being the point of X_j closest to $\boldsymbol{x}_{i,k}$, a two dimensional rotation $\boldsymbol{R}_\beta$ with β being the rotation angle and a translation vector $\boldsymbol{t} = [t_x, t_y]^\top$ can be calculated. The loop closing constraint can then be derived using Equation (1) by transforming $\hat{\boldsymbol{p}}_j$ given the rotation and translation which leads to

$$\hat{\boldsymbol{\xi}}_{ij} = \begin{bmatrix} t_x & t_y & \beta \end{bmatrix}^\top . \tag{10}$$

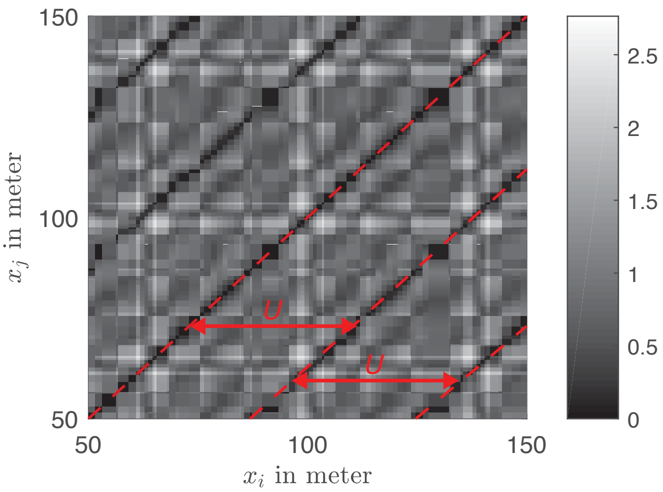


Fig. 3: Comparison error of the shapes of the neighborhood between the vertices of the pose graph. For better reading we plotted the error in the form $\log(1 - \boldsymbol{C}_{ij})$ and only a section of the matrix. The variables x_i and x_j are representing the position l of the vertices i and j in meter along the pose graph. The estimated circumference U for the closed environment can be read directly from the graphic.

The corresponding covariance matrix can be calculated using the correlation error C_{ij} and tuneable parameters γ_1 and γ_2

$$\boldsymbol{P}_{\text{lc},ij} = \text{diag}\left(\begin{bmatrix}\gamma_1 & \gamma_1 & \gamma_2\end{bmatrix}\right) C_{ij}. \tag{11}$$

The parameters γ_1 and γ_2 are constant for all loop closing constraints and will be learned through Bayesian Optimization. This process is discussed in Section III-B. For the odometric constraints we generate the covariance matrices as

$$\boldsymbol{P}_{\text{odometric},ij} = \text{diag}\left(\begin{bmatrix}\cos(\varphi_i)(\alpha_3\delta_T + \alpha_4\delta_R) \\ \sin(\varphi_i)(\alpha_3\delta_T + \alpha_4\delta_R) \\ \alpha_1\delta_R + \alpha_2\delta_T\end{bmatrix}\right) \tag{12}$$

on the basis of the odometry model presented in [32] and under the assumption that only one translation δ_T and one rotation δ_R occur. The parameters $\alpha_1, \ldots, \alpha_4$ can be learned. Here, we assume these parameters are given due to a known odometry model of the underlying differential drive system.

C. Recurrent Symmetric Structures

A problem for the approach introduced above are recurrent symmetric structures. Such structures are present in many real world scenarios, and hence an autonomous robot needs to be able to cope with them. Therefore, we introduce a feasibility check

$$|\pi - \text{mod}\left(\Delta\varphi_{ij}, 2\pi\right)| > \varphi_{\text{cycle}} \tag{13}$$

for every loop closing pair $\{\boldsymbol{p}_i, \boldsymbol{p}_j\}$ with the respective difference in orientation $\Delta\varphi_{ij} = \varphi_j - \varphi_i$. Here, $\text{mod}(a, b)$ is the modulo function which gives back the remainder of the Euclidean division of a by b. Only loop closing pairs which pass the check of Equation (13) are considered for pose graph optimization. The feasibility check is based on the assumption, that, on average, the orientation error of the odometric measurements will sum up to zero. However, the orientation difference can largely differ and thus the meta-parameter $\varphi_{\text{cycle}} \in [0, \pi]$ has to be selected accordingly.

III. Meta-Parameter Learning

To learn the unknown meta-parameters for the above mapping algorithm, we define an optimization problem with the objective

$$\min_{\boldsymbol{\theta}} c(\boldsymbol{\theta}) \tag{14}$$

as a general cost function. This cost is then minimized through episodic BO [29] with expected improvement [23]. To optimize both terms, the loop closing parameters L_{NH}, c_{min}, φ_{cycle} and the pose graph optimization parameters γ_1, γ_2 we define a two-stage optimization process. First, we optimize the loop closing parameters L_{NH}, c_{min}, φ_{cycle} which gives us as a by-product an estimate of the circumference of the closed environment U. Based on the estimated circumference U we can define a cost function for optimizing the pose graph parameters γ_1, γ_2. Hence, a joint optimization of all parameters is not suitable. In the following, we derive the two cost functions required for the optimization process.

A. Stage 1 – Optimization of Loop Closing Parameters

We assume that the odometric error between two poses i and j is on average zero. This is a quite strong assumption, however, a non-zero mean value will be inherent in the generated map and thus compensated when navigating with the same robot odometry. To model this error, we use a Gaussian Distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{P})$ with the covariance matrix $\boldsymbol{P}$. Let u then denote the distance along the pose graph between a loop closing pair i, j

$$u = \sum_{k=i}^{j-1} ||\boldsymbol{x}_{k+1} - \boldsymbol{x}_k||. \tag{15}$$

Given the assumption from above, the path distances for all loop closing pairs $\boldsymbol{u} = [u_1, u_2, \ldots, u_M]$, identified by cycling around a closed environment, are, on average, multiples of the circumference nU. Here, $n \in \mathbb{N}^+$ is a positive integer, representing the number of cycles before the loop closure detection. Hence, if all loop closures are detected properly, a histogram of the path distances $\boldsymbol{u}$ has only equally distributed peaks at positions nU. The right panels of Figure 4 show such histograms for ill-detected loop closures (top) and well-detected loop closures (bottom). To transform this idea into a cost function, we can learn a Gaussian Mixture Model (GMM) [27] with the probability distribution

$$p(\boldsymbol{u}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{u}|\mu_k, \Sigma_k) \tag{16}$$

from observed path distances $\boldsymbol{u}$. Here, K is the number of mixture components and π_k, μ_k, Σ_k the mixture weight, the mean and the variance of the k-th component respectively. As part of the cost function we use the negative log likelihood of the GMM

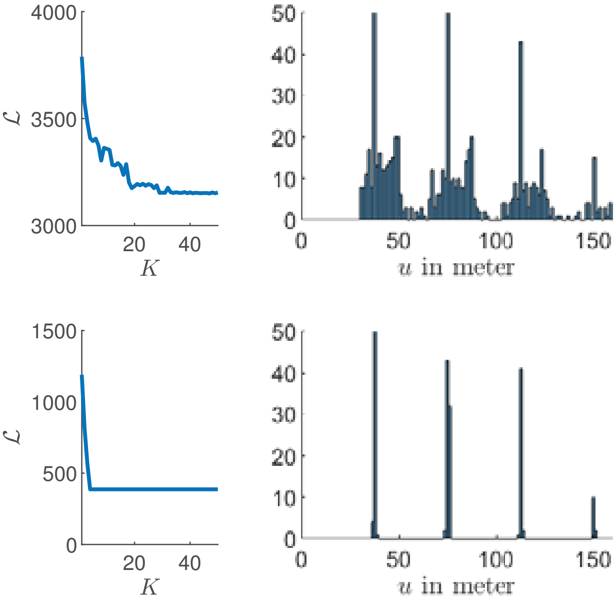


Fig. 4: For the top panels, the mapping parameters have been ill-chosen. The upper left panel shows the negative log likelihood history and the upper right panel the histogram for the path distances of the loop closing pairs. In the bottom panels the mapping parameters have been well-chosen. Again, in the left panel the negative log likelihood history is shown and in the right the histogram.

$$-\mathcal{L} = -\ln p(\boldsymbol{u}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\sum_{i=1}^{M} \ln \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(u_i|\mu_k, \Sigma_k)\right] \tag{17}$$

over the data set $\boldsymbol{u} = [u_1, u_2, \ldots, u_M]$. The log likelihood decreases if the dataset $\boldsymbol{u}$ meets the above assumption of evenly distributed peaks at positions nU. A common strategy for training GMMs is to iteratively increasing K until the log likelihood does not improve further. In the left column in Figure 4, the evolution of the negative log likelihood with respect to the number of components of the GMM is shown. For fitting the GMM the iterative Expectation-Maximization (EM) algorithm is used [8], [22]. The EM algorithm starts with a randomly selected model and then alternately optimizes the allocation of the data $\boldsymbol{u}$, i.e. the weighting π_k, to the individual parts of the model and the parameters of the model μ_k and Σ_k. If there is no significant improvement, the procedure is terminated.

We define the cost function for the loop closure detection as

$$\min_{\boldsymbol{\theta}} c(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} (-\mathcal{L} - \log(M)), \tag{18}$$

with the unknown parameters $\boldsymbol{\theta} = [L_{\text{NH}}, c_{\text{min}}]$, the length of the neighborhood and the minimum comparison error, and M being the number of loop closures found. The cost function represents a trade-off between the number of loop closures, where more reliable loop closures result in a better pose graph optimization, and a restrictive choice of loop closures to avoid false detection. Based on the best GMM fit, the circumference of the closed environment U can be estimated.

To also learn the meta-parameter φ_{cycle} for the feasibility check for recurrent symmetric structures, we can calculate the negative log likelihood for orientation differences of the loop closing pairs $\Delta\varphi_{ij}$, similar to Equation (17), under the assumption that all $\Delta\varphi_{ij}$ for accurate loop closures are close to $n2\pi$ with $n \in \mathbb{N}^{+}$. Equation (18) than turns into

$$\min_{\boldsymbol{\theta}} c(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \left(-\mathcal{L} - \mathcal{L}_{\varphi} - \log(M)\right), \tag{19}$$

with $\boldsymbol{\theta}$ being now $\boldsymbol{\theta} = [L_{\text{NH}}, c_{\text{min}}, \varphi_{\text{cycle}}]$. The additional cost term $-\mathcal{L}_{\varphi}$ represents the negative log likelihood of the GMM from Equation (17) over the data set $\Delta\boldsymbol{\varphi} = \{\Delta\varphi_1, \Delta\varphi_2, \ldots, \Delta\varphi_M\}$, thus

$$-\mathcal{L}_{\varphi} = -\ln p(\Delta\boldsymbol{\varphi}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{20}$$

B. Stage 2 – Optimization of Pose Graph Parameters

Based on our assumption of a zero mean odometric error we can assume the estimated circumference U from the first stage of our optimization process to be the true circumference of the closed environment. Hence, we can define a cost function for learning the pose graph optimization parameters $\boldsymbol{\gamma} = [\gamma_1, \gamma_2]$ as

$$\min_{\boldsymbol{\gamma}} c(\boldsymbol{\gamma}) = \min_{\boldsymbol{\gamma}} |U - \hat{U}| \tag{21}$$

where $\hat{U}$ represents the estimated circumference after pose graph optimization. Thus, we punish deviations between the estimated circumference based on the original pose graph and the optimized one. In order to estimate the circumference after pose graph optimization, a fit onto GMMs is performed as proposed in Section III-A.

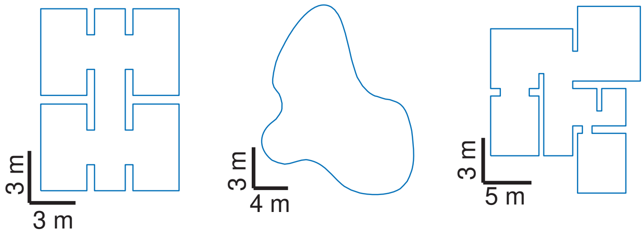
IV. Results

We evaluated the accuracy of the pose graph optimization (performance) and the generality of our approach in different environments (robustness). As a measure for the performance, we used an error metric based on the relative displacement between poses

$$E_{\text{rel}}(\boldsymbol{\xi}) = \frac{1}{N}\sum_{i,j} \text{trans}\left(\boldsymbol{\xi}_{i,j} \ominus \boldsymbol{\xi}^{*}_{i,j}\right)^2 + \text{rot}\left(\boldsymbol{\xi}_{i,j} \ominus \boldsymbol{\xi}^{*}_{i,j}\right)^2 \tag{22}$$

as introduced in [5]. Here, $\boldsymbol{\xi}_{i,j}$ are the relative transformations after pose graph optimization, $\boldsymbol{\xi}^{*}_{i,j}$ ideally the true relative transformations and *trans* and *rot* separate the translational and rotational components. Additionally, we used a second error metric for comparing results obtained on real lawns where the true poses of the robot are unknown but a groundtruth of the environment is available. Therefore, we constructed a polygon defined by the points X out of the optimized pose graph data and compare this polygon with a polygon representing the groundtruth, X_{true}. We then transform

$$X \leftarrow \boldsymbol{R} \cdot X + \boldsymbol{t}, \tag{23}$$



(a) Map 1 (b) Map 2 (c) Map 3

Fig. 5: Simulation environments used for evaluating the proposed learning procedure for mapping in closed environments. From left to right: A symmetric environment ($U = 77\,m$), a curved environment ($U = 52\,m$) and an apartment environment ($U = 100\,m$).

such that the deviation between the enclosed areas A, A_{true} of the polygons

$$\Delta A = 1 - \frac{A_{\text{true}} \cap A_{\text{estimate}}}{A_{\text{true}} \cup A_{\text{estimate}}} \tag{24}$$

is minimized. Here, $\boldsymbol{R}$ is a rotation matrix and $\boldsymbol{t}$ a translational vector. The minimized difference then serves as secondary error metric.

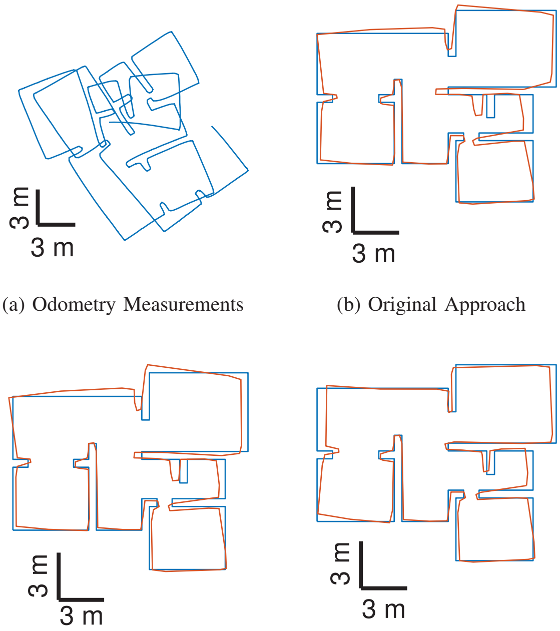
We compared to the original approach from [28] using hand crafted and learned parameters. The handcrafted parameters have been selected according to the following rules:
The neighborhood L_{NH} should be chosen such that $2L_{\text{NH}}$ is slightly larger then half of the true circumference. Thus, we like to use slightly more than $50\,\%$ of U for shape comparison. The comparison error threshold c_{min} should be chosen according to the complexity of the given map. A more complex map requires a larger comparison error threshold to account for more complicated comparisons. The meta-parameter φ_{cycle} has to be chosen with regard to the recurrent symmetric structures of the given map. Here, only Map 1 has such structures with which we can cope by setting $\varphi_{\text{cycle}} = \pi/2$. The pose graph optimization parameters γ_1, γ_2 are kept constant with $\gamma_1 = 1$ and $\gamma_2 = 1$.

A. Simulation

We show the robustness of the approach applying our mapping procedure in different simulated closed environments with hard features, such as recurrent structures, large dimensions or curvatures. For the simulation environment, we used the odometry motion model presented in [32]. We calibrated the odometry model by tracking lawn mower movements using a visual tracking system (OptiTrack) and computed the parameters using maximum likelihood estimation [25]. The calibrated parameters for the Viking MI 422P robot are presented in Table II and are used for the simulation. To generate movement data, we used a wall-following algorithm cycling for $T = 2000\,s$ along the boundary of the closed environment. We statistically evaluated our approach

TABLE II: Measured Parameters for the odometry motion model [32].

α_1	α_2	α_3	α_4
0.0849	0.0412	0.0316	0.0173



(a) Odometry Measurements

(b) Original Approach

(c) Adjusted Approach

(d) Learned Parameters

Fig. 6: Exemplary mapping results with the simulation environment ”Map 3” and an odometry error of $\alpha_i = 0.2$. In (b)-(d) the blue line shows the true shape of the environment and the red line the map estimate.

simulating 20 runs with a maximum of 30 iterations for the Bayes Optimizer. This optimization is designed for global optimization of black-box functions and does not require any derivatives.

In Table III, the simulation results for the different maps from Figure 5 with different combinations of hand-crafted parameters are presented. Our adjusted approach, using the ICP method, clearly outperforms the original method. In all 20 runs it leads to better map estimates after pose graph optimization. Moreover, learning the parameters enables the algorithm to generalize to different environments without prior knowledge about the odometry error, the shape or the circumference of the environment. This prior knowledge is essential for choosing suitable hand-crafted parameters. Without such knowledge, the parameters have to be manually tuned which might lead to disastrous mapping results. For example, changing the neighborhood parameter L_{NH} for Map 3 to $L_{\text{NH}} = 15$ results in a large increase of the mapping errors. In addition, a change in odometry error accuracy can be compensated by learning the mapping meta-parameters, as demonstrated in simulations with odometry model parameters $\alpha_i = 0.1, 0.2$ for $i = 1, \dots, 4$.

B. *Real Data*

For generating real data, we drove the lawn mower along the boundary line of two different lawn areas. The velocity of the lawn mower driving along the boundary has been set to $0.3\,\mathrm{m\,s^{-1}}$. The odometry data has been sampled with a frequency of approximately $20\,\mathrm{Hz}$.

In Figure 7, the university courtyard, the measured odometry data and the generated map estimate are shown. The ground truth is available as CAD data, such that we can compare our map estimations using Equation (24). Based on the circumference $U = 106.8\,m$ and the complexity of the environment, the hand crafted parameters have been set to $L_{\text{NH}} = 30$, $c_{\text{min}} = 0.3$. The resulting mapping error for the original approach is $\Delta A = 11.49\%$, for the adjusted approach $\Delta A = 9.77\%$ and the mapping error with learned parameters $c_{\text{min}} = 0.1967$, $L_{\text{NH}} = 32.32$, $\varphi_{\text{cycle}} = 1.57$, $\gamma_1 = 0.0104$, $\gamma_2 = 0.0122$ is $\Delta A = 9.24\%$. Again, the adjusted approach outperforms the original approach and learning the parameters with the proposed cost functions leads to sufficiently accurate results.

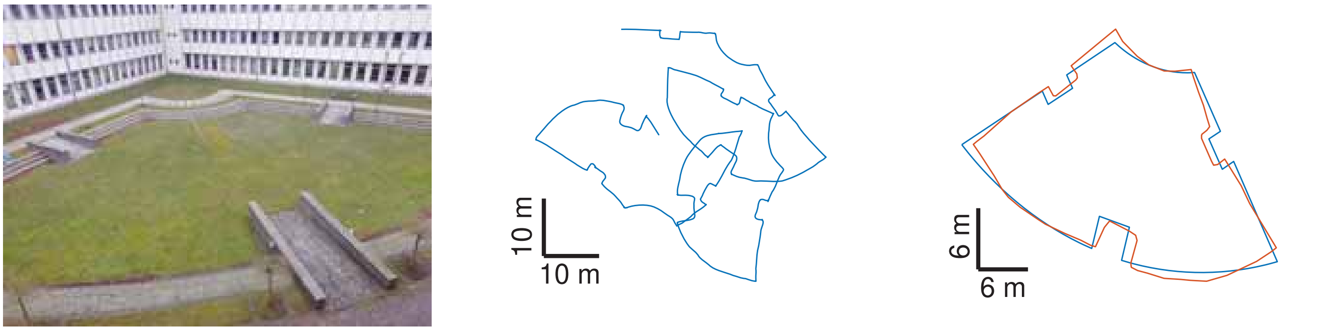
In addition, we evaluated the mapping approach in a second real environment, a representative of a typical private lawn. In Figure 8 from left to right, we show a part of the private lawn, the measured odometry data and the map estimate. Since we do not have ground truth data for this lawn, we compared the map results qualitatively with the image of the real garden. As demonstrated, the approach is capable of mapping large closed environments with narrow corridors based on severely distorted odometry data.

V. Conclusion

Towards efficient localization and planning for low-cost robots, a first step is the generation of an accurate map estimate of the enclosed environment. Thereby, the robot has to learn required meta-parameters automatically to be able to adapt to different environments. Here, we have made improvements to the mapping algorithms for closed environment introduced in [28], which significantly enhance the performance by allowing the algorithm to cope with recurrent symmetric structures as well as reducing the relative displacement error. Moreover, we proposed a cost function for meta-parameter learning for mapping algorithms in closed environments. This cost function does neither require any a-priori information about the environment nor domain expert knowledge and thus enables the robot to act truly autonomously. We demonstrated the feasibility, robustness and performance of our approach in both simulated and real closed environments. Thereby, we showed that based on the proposed mapping procedure, accurate map estimates of underlying closed environments can be produced. These map estimates are the first step towards intelligent behavior for low-cost robots, such as autonomous lawn mowers.

TABLE III: Simulation results for different maps and hand-crafted parameters for the original approach from [28], the adapted approach and with learned parameters. The table shows the mean and the standard deviations for the relative displacement errors.
*the measured odometry parameters from Table II are used

Map	L_{NH}	c_{min}	α_i	Original Approach		Adjusted Approach		Learned Parameters	
				E_{trans}	E_{rot}	E_{trans}	E_{rot}	E_{trans}	E_{rot}
1	20	1.0	*	0.0036 ± 0.0028	0.0076 ± 0.0053	$\mathbf{0.0021} \pm 0.0034$	$\mathbf{0.0052} \pm 0.0055$	0.0006 ± 0.0015	0.0093 ± 0.0166
2	15	0.5	*	0.0406 ± 0.0943	0.0569 ± 0.0399	0.0183 ± 0.0068	0.0564 ± 0.0404	$\mathbf{0.0002} \pm 0.0005$	$\mathbf{0.0003} \pm 0.0003$
3	30	0.3	*	0.1290 ± 0.5532	0.0049 ± 0.0098	$\mathbf{0.0020} \pm 0.0050$	$\mathbf{0.0021} \pm 0.0021$	0.0024 ± 0.0036	0.0262 ± 0.0780
3	30	1.5	*	2.787 ± 12.16	0.0095 ± 0.0093	0.5442 ± 2.004	0.0063 ± 0.0136	$\mathbf{0.0017} \pm 0.0026$	$\mathbf{0.0335} \pm 0.0607$
3	15	0.3	*	35.57 ± 158.0	0.0191 ± 0.0214	31.03 ± 137.6	0.0139 ± 0.0220	–	–
3	30	0.3	0.1	0.0151 ± 0.0418	0.0060 ± 0.0086	0.0085 ± 0.0296	0.0025 ± 0.0020	$\mathbf{0.0070} \pm 0.0079$	$\mathbf{0.0665} \pm 0.1932$
3	30	0.3	0.2	44.54 ± 188.8	0.0304 ± 0.0580	1.65 ± 6.88	0.0158 ± 0.0285	$\mathbf{0.0205} \pm 0.0387$	$\mathbf{0.0352} \pm 0.0642$

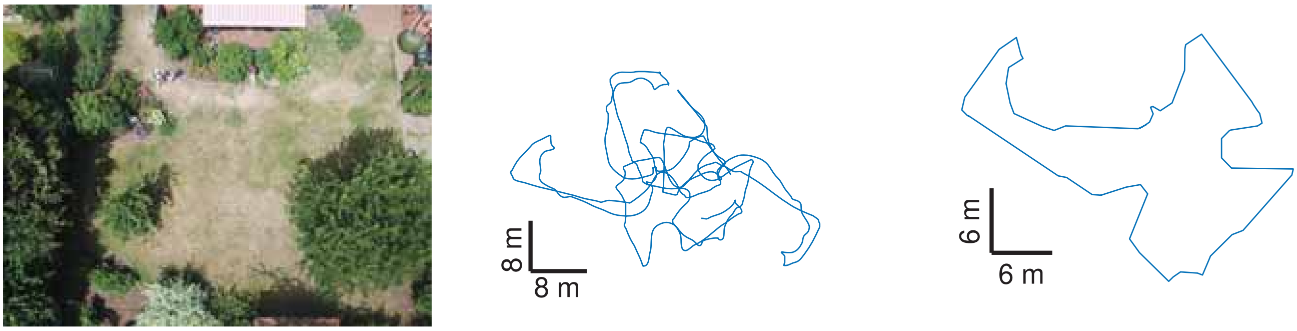


(a) The courtyard of our Institute. We used the inner lawn area for testing the proposed mapping method.

(b) The estimated path of the robot generated from its wheel odometry.

(c) The estimated map (red) and the true shape of the test environment (blue).

Fig. 7: The real courtyard depicted (a), the collected odometry data (b) and the map estimate with learned parameters (c).



(a) The top view onto a part of a lawn of a typical private household.

(b) The estimated path of the robot generated from its wheel odometry.

(c) The estimated map.

Fig. 8: A typical lawn (a), the collected odometry data (b) and the map estimate with learned parameters (c).

A. *Discussion*

The underlying assumption of a zero mean odometry error is quite strong and might not hold true under many circumstances, for example if one of the wheels is slightly smaller (e.g. due to air pressure). However, fusing the wheel odometry with IMU measurements, we are able to compensate for such inaccuracies. Moreover, we can detect wheel slippage. Otherwise, a non-zero odometric mean error will be inherited in the final map estimate and thus compensated by navigating with the same robot odometry.

In future work, we will investigate the possibilities of probabilistic approaches for efficiently mowing the lawn with high-confidence. Therefore, coverage grid maps with ”already mown lawn” probabilities similar as in [14] can be used in combination with an adjusted intelligent complete coverage path planning algorithm, e.g. neural network approach [33]. Thereby, the ”mowing probabilities” of the grid map are actualized based on a particle filter estimation.

References

[1] Yassin Abdelrasoul, Abu Bakar Sayuti HM Saman, and Patrick Sebastian. A quantitative study of tuning ros gmapping parameters and their effect on performing indoor 2d slam. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 1–6. IEEE, 2016.

[2] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.

[3] Per Bergström and Ove Edlund. Robust registration of point sets using iteratively reweighted least squares. *Computational optimization and applications*, 58(3):543–561, 2014.

[4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[5] Wolfram Burgard, Cyrill Stachniss, Giorgio Grisetti, Bastian Steder, Rainer Kümmerle, Christian Dornhege, Michael Ruhnke, Alexander Kleiner, and Juan D Tardós. A comparison of slam algorithms based on a graph of relations. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2089–2095. IEEE, 2009.

[6] Young-Ho Choi, Tae-Kyeong Lee, and Se-Young Oh. A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot. *Autonomous Robots*, 24(1):13–27, 2008.

[7] V Ciupe and I Maniu. New trends in service robotics. In *New trends in medical and service robots*, pages 57–74. Springer, 2014.

[8] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[9] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[10] Nils Einecke, Jörg Deigmöller, Keiji Muro, and Mathias Franzius. Boundary wire mapping on autonomous lawn mowers. In *Field and Service Robotics*, pages 351–365. Springer, 2018.

[11] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[12] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2432–2437. IEEE, 2005.

[13] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard, et al. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34, 2007.

[14] Jürgen Hess, Maximilian Beinhofer, and Wolfram Burgard. A probabilistic approach to high-confidence cleaning guarantees for low-cost cleaning robots. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 5600–5605. IEEE, 2014.

[15] Martin Hägele. Robots conquer the world [turning point]. *IEEE Robotics & Automation Magazine*, 23(1):120–118, 2016.

[16] Myung-Jin Jung, Hyun Myung, Sun-Gi Hong, Dong-Ryeol Park, Hyoung-Ki Lee, and SeokWon Bang. Structured light 2d range finder for simultaneous localization and map-building (slam) in home environments. In *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004.*, pages 371–376. IEEE, 2004.

[17] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.

[18] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.

[19] Hyoung-Ki Lee, Kiwan Choi, Jiyoung Park, and Hyun Myung. Self-calibration of gyro using monocular slam for an indoor mobile robot. *International Journal of Control, Automation and Systems*, 10(3):558–566, 2012.

[20] Seongsoo Lee, Sukhan Lee, and Seungmin Baek. Vision-based kidnap recovery with slam for home cleaning robots. *Journal of Intelligent & Robotic Systems*, 67(1):7–24, 2012.

[21] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

[22] Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. M. Dekker New York, 1988.

[23] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.

[24] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002.

[25] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[26] Ozan Ozisik and Sirma Yavuz. Simultaneous localization and mapping with limited sensing using extended kalman filter and hough transform. *Tehnicki vjesnik/Technical Gazette*, 23(6), 2016.

[27] Douglas Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, pages 827–832, 2015.

[28] Nils Rottmann, Ralf Bruder, Achim Schweikard, and Elmar Rueckert. Loop closure detection in closed environments. In *European Conference on Mobile Robots*, pages 1–8, 2019. https://arxiv.org/abs/1908.04558.

[29] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[30] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[31] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.

[32] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[33] Simon X Yang and Chaomin Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):718–724, 2004.

[34] Ying Zhang, Juan Liu, Gabriel Hoffmann, Mark Quilling, Kenneth Payne, Prasanta Bose, and Andrew Zimdars. Real-time indoor mapping for mobile robots with limited sensing. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pages 636–641. IEEE, 2010.

Radar-Camera Sensor Fusion for Joint Object Detection and Distance Estimation in Autonomous Vehicles

Ramin Nabati[1] and Hairong Qi[1]

Abstract— **In this paper we present a novel radar-camera sensor fusion framework for accurate object detection and distance estimation in autonomous driving scenarios. The proposed architecture uses a middle-fusion approach to fuse the radar point clouds and RGB images. Our radar object proposal network uses radar point clouds to generate 3D proposals from a set of 3D prior boxes. These proposals are mapped to the image and fed into a Radar Proposal Refinement (RPR) network for objectness score prediction and box refinement. The RPR network utilizes both radar information and image feature maps to generate accurate object proposals and distance estimations.**

The radar-based proposals are combined with image-based proposals generated by a modified Region Proposal Network (RPN). The RPN has a distance regression layer for estimating distance for every generated proposal. The radar-based and image-based proposals are merged and used in the next stage for object classification. Experiments on the challenging nuScenes dataset show our method outperforms other existing radar-camera fusion methods in the 2D object detection task while at the same time accurately estimates objects' distances.

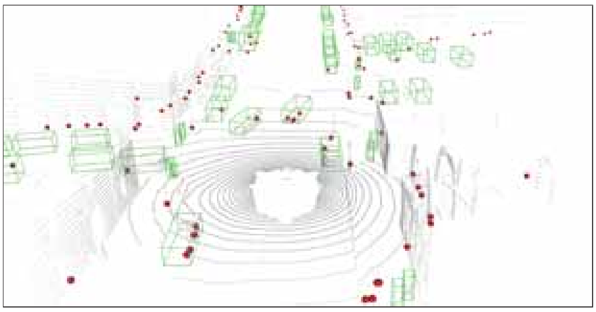
I. INTRODUCTION

Object detection and depth estimation is a crucial part of the perception system in autonomous vehicles. Modern self driving cars are usually equipped with multiple perception sensors such as cameras, radars and LIDARs. Using multiple sensor modalities provides an opportunity to exploit their complementary properties. Nonetheless, the process of multi-modality fusion also makes designing the perception system more challenging. Over the past few years many sensor fusion methods have been proposed for autonomous driving applications. Most existing sensor fusion algorithms focus on combining RGB images with 3D LIDAR point clouds [1]. LIDARs provide accurate depth information that could be used for 3D object detection. This is particularly useful in autonomous driving applications where having the distance to all detected objects is crucial for safe operation.

While LIDARs are becoming popular in autonomous vehicles, radars have been used in autonomous and also non-autonomous vehicles for many years as an indispensable depth sensor. Radars operate by measuring the reflection of radio waves from objects, and use the Doppler effect to estimate objects' velocity. Although radars provide accurate distance and velocity information, they are not particularly good at classifying objects. This makes the fusion of radar and other sensors such as cameras a very interesting topic in autonomous driving applications. A radar-camera fusion system can provide valuable depth information for all detected objects in an autonomous driving scenario, while at the same time eliminates the need for computationally expensive 3D object detection using LIDAR point clouds.

Due to their unstructured nature, processing depth sensor data is a very challenging problem. Additionally, the point cloud obtained by depth sensors are usually sparse with very variable point density. In LIDAR point clouds for example, nearby objects have significantly more measurements than far away objects. This makes the point cloud-based object detection a challenging task. To overcome this problem, some methods apply image-based feature extraction techniques by projecting the point cloud into a perspective view [2], [3], [4], e.g. the bird's eye view (BEV). Other methods [4], [5], [6] partition the point cloud into a regular grid of equally spaced voxels, and then learn and extract voxel-level features. More recently, Qi *et al.* [7], [8] proposed PointNet, an end-to-end deep neural network for learning point-wise features directly from point clouds for segmentation and classification.

Although point cloud feature extraction and classification methods have proven to be very effective on dense point clouds obtained from LIDARs, they are not as effective on sparse radar point clouds. For one object, an ideal radar only reports one point, compared to tens or hundreds of points obtained by a LIDAR for the same object. Additionally, most automotive radars do not provide any height information for the detected objects, essentially making the radar point clouds a 2-dimensional signal, as opposed to the 3-dimensional point clouds obtained from a LIDAR. Another difference between radar and LIDAR point clouds



(a)

Fig. 1: Sample data from the NuScenes dataset showing Radar point cloud (red), 3D ground truth boxes (green) and LIDAR point cloud (grey).

[1]Department of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, USA. Email: mnabati@vols.utk.edu, hqi@utk.edu

is the amount of processing needed to extract useful features. Automotive radars have built-in functionalities to extract very useful features for every detection, such as relative object speed, detection validity probability and stationary or moving classification for objects. While one can use these features directly without any further processing, LIDAR point clouds require extensive processing to obtain object-level features. These differences make processing radar point clouds different and sometimes more challenging compared to LIDAR point clouds.

Some existing point-based proposal generation methods process point cloud by first projecting it to different views or using voxels to represent it in a compact form. 2D or 3D convolutional networks are then used to extract features. Other methods extract features from the raw point clouds directly using networks such as PointNet [8]. These methods are usually designed for dense LIDAR point clouds and do not perform equally well on sparse radar point clouds. Additionally, unlike LIDAR point clouds, radar point clouds do not provide a precise 3D image of the object, as an ideal radar reports only one point for an object. Aggregating multiple radar readings obtained in different time-stamps can help provide more points in the point cloud, but these points are not a good representation of the objects' shape and size. Fig. 1 visualizes some of these differences by showing radar and LIDAR point clouds for a sample scene from the nuScenes dataset.

In this work, we propose a radar-camera fusion algorithm for joint object detection and distance estimation in autonomous driving applications. The proposed method is designed as a two-stage object detection network that fuses radar point clouds and learned image features to generate accurate object proposals. For every object proposal, a depth value is also calculated to estimate the object's distance from the vehicle. These proposals are then fed into the second stage of the detection network for object classification. We evaluate our network on the nuScenes dataset [9], which provides synchronized data from multiple radar and camera sensors on a vehicle. Our experiments show that the proposed method outperforms other radar-camera fusion methods in the object detection task and is capable of accurately estimating distance for all detected objects.

II. Related work

In this section we highlight some of the existing works on object detection and sensor fusion for autonomous vehicles, categorizing them into single-modality and fusion-based approaches.

A. Single-Modality Object Detection

Most vision-based object detection networks follow one of the two approaches: two-stage or single-stage detection pipelines [10]. In two-stage detection networks, a set of class-agnostic object proposals are generated in the first stage, and are refined, classified and scored in the second stage. R-CNN [11] is the pioneering work in this category, using proposal generation algorithms such as Selective Search [12] in the first stage and a CNN-based detector in the second stage. Fast R-CNN [13] also uses an external proposal generator, but eliminates redundant feature extraction by utilizing the global features extracted from the entire image to classify each proposal in the second stage. Faster R-CNN [14] unifies the proposal generation and classification by introducing the Region Proposal Network (RPN), which uses the global features extracted from the image to generate object proposals.

One-stage object detection networks on the other hand directly map the extracted features to bounding boxes by treating the object detection task as a regression problem. YOLO [15] and SSD [16] detection networks are in this category, regressing bounding boxes directly from the extracted feature maps. One-stage detection networks are usually faster, but less accurate than their two-stage counterparts. By addressing the foreground-background class imbalance problem in single-stage object detection, RetinaNet [17] achieved better results than the state-of-the-art two-stage detection networks.

Most of the point-based object detection networks focus on dense point clouds obtained from LIDARs. Some of these methods process the points by discretizing the 3D space into 3D voxels [18], [19], while others process the point clouds in the continuous vector space without voxelization to obtain individual features for each point [7], [8]. For object detection and classification using radar data, [20] proposes radar grid maps by accumulating radar data over several time-stamps, while [21] uses CNNs on a post-processed range-velocity map. The radar data can also be processed as a 3D point cloud. [22] and [23] both use PointNet to perform 2D object classification and segmentation, respectively.

B. Fusion-based Object Detection

Most fusion-based methods combine the LIDAR point clouds with RGB images for 2D or 3D object detection [24], [25]. In [2] the network uses a multi-view representation of the 3D LIDAR point clouds. The network projects the points to the Bird's Eye View (BEV) and front view planes, and uses the BEV to generate object proposals. [26] projects radar detections to the image and generate object proposals for a small CNN classification network. In [27], authors map radar detection to the image plane and use a radar-based RPN to generate 2D object proposals for different object categories in a two-stage object detection network. Authors in [28] also project radar detections to the image plane, but represent radar detection characteristics as pixel values. The RGB image is then augmented with these values and processed in a CNN to regress 2D bounding box coordinates and classification scores.

III. Our Framework

Our proposed sensor fusion network is shown in Fig. 2. The network takes radar point clouds and RGB images as input and generates accurate object proposals for a two-stage object detection framework. We take a middle-fusion approach for fusing the radar and image data, where outputs of

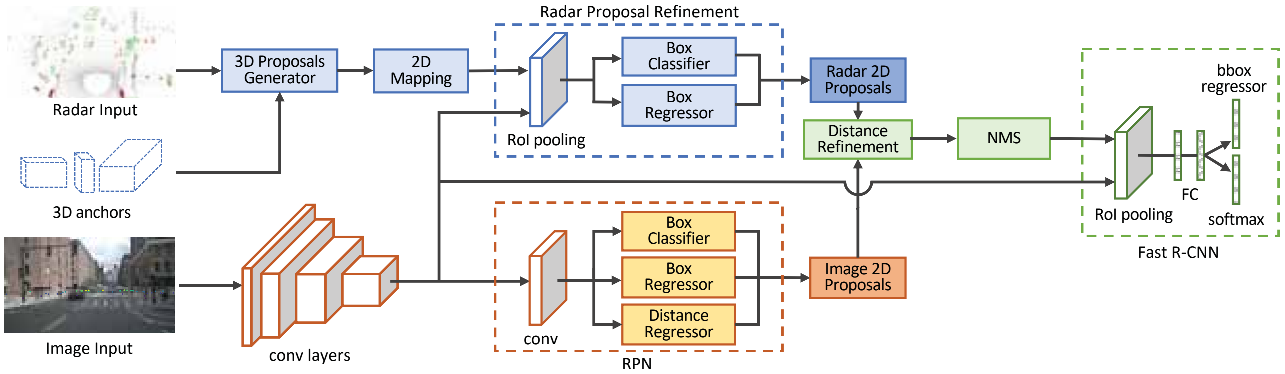


Fig. 2: The proposed network architecture. Inputs to the network are radar point cloud, camera image and 3D anchor boxes. radar-based object proposals are generated from the point cloud and fused with image features to improve box localization.

each sensor are processed independently first, and are merged at a later stage for more processing. More specifically, we first use the radar detections to generate 3D object proposals, then map the proposals to the image and use the image features extracted by a backbone network to improve their localization. These proposals are then merged with image-based proposals generated in a RPN, and are fed to the second stage for classification. All generated proposals are associated with an estimated depth, calculated either directly from the radar detections, or via a distance regressor layer in the RPN network.

A. *Radar Proposal Network*

Our proposed architecture treats every radar point as a stand-alone detection and generates 3D object proposals for them directly without any feature extraction. These proposals are generated using predefined 3D anchors for every object class in the dataset. Each 3D anchor is parameterized as (x, y, z, w, l, h, r), where (x, y, z) is the center, (w, l, h) is the size, and (r) is the orientation of the box in vehicle's coordinate system. The anchor size, (w, l, h), is fixed for each object category, and is set to the average size of the objects in each category in the training dataset. For every anchor box, we use two different orientations, r = $\{0°, 90°\}$ from the vehicle's centerline. The center location for each anchor is obtained from the radar detection's position in the vehicle coordinates. For every radar point, we generate *2n* boxes from the 3D anchors, where *n* is the number of object classes in the dataset, each having two different orientations.

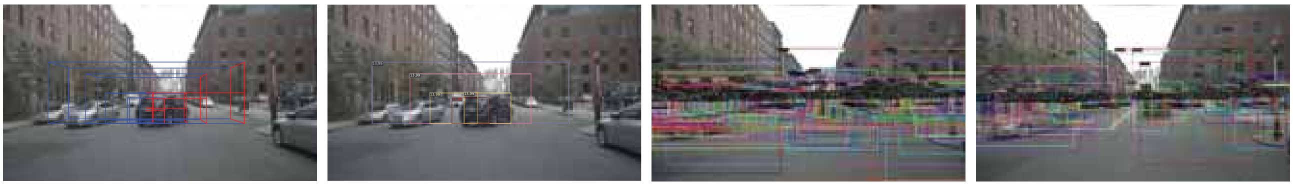
In the next step, all 3D anchors are mapped to the image plane and converted to equivalent 2D bounding boxes by finding the smallest enclosing box for each mapped anchor. Since every 3D proposal is generated from a radar detection, it has an accurate distance associated with it. This distance is used as the proposed distance for the generated 2D bounding box. Since 3D anchors with the same size as objects of interest are used to generate the 2D object proposals on the image, the resulting proposals capture the true size of the objects as they appear in the image. This eliminates the need for adjusting the size of radar proposals based on their distance from the vehicle, which was proposed in [27].

Fig. 3(b) illustrates 3D anchors and equivalent 2D proposals generated for a sample image. As shown in this figure, radar-based proposals are always focused on objects that are on the road plane. This prevents unnecessary processing of areas of the image where no physical object exists, such as the sky or buildings in this image.

In the next step, all generated 2D proposals are fed into the Radar Proposal Refinement (RPR) subnetwork. This is where the information obtained from the radars (radar proposals) is fused with the information obtained from the camera (image features). RPR uses the features extracted from the image by the backbone network to adjust the size and location of the radar proposals on the image. As radar detections are not always centered on the corresponding objects on the image, the generated 3D anchors and corresponding 2D proposals might be offset as well. The box regressor layer in the RPR uses the image features inside each radar proposal to regress offset values for the proposal corner points. The RPR also contains a box classification layer, which estimates an objectness score for every radar proposal. The objectness score is used to eliminate proposals that are generated by radar detections coming from background objects, such as buildings and light poles. The inputs to the box regressor and classifier layers are image features inside negative and positive radar proposals. We follow [14] and define positive proposals as ones with an Intersection-over-Union (IoU) overlap higher than 0.7 with any ground truth bounding box, and negative proposals as ones with an IoU below 0.3 for all ground truth boxes. Radar proposals with an IoU between 0.3 and 0.7 are not used for training. Since radar proposals have different sizes depending on their distance, object category and orientation, a RoI Pooling layer is used before the box regression and classification layers to obtain feature vectors of the same size for all proposals. Fig. 3(d) shows the radar proposals after the refinement step.

B. *Image Proposal Network*

Our architecture also uses a RPN network to generate object proposals from the image. The radar proposal network is not always successful in generating proposals for certain object categories that are harder for radars to detect but are



(a) (b) (c) (d)

Fig. 3: Radar-based proposals. (a): 3D anchors for one radar detection ($r = 90°$). (b): 2D proposals obtained from 3D anchors. (c): 2D proposals for all radar detections inside the image. (d): Refined radar proposals after applying box regression. Radar-based distances in meters are shown on the bounding boxes.

easily detected in the image, such as pedestrian or bicycles. On the other hand, the image-based proposal network might fail to detect far away objects that are easily detected by the radar. Having an image-based object proposal network in addition to the radar-based network improves the object detection accuracy, as they complement each other by using two different modalities for proposal generation and distance estimation.

Image-based object proposals are generated by a network similar to the RPN introduced in Faster R-CNN [14]. The input to this network is the image feature maps extracted by the backbone CNN. To estimate distance for every object proposal, we add a fully connected distance regression layer on top of the convolutional layer in RPN, as shown in Fig. 2. This layer is implemented with a 1×1 convolutional layer similar to the box-regression and box-classification layers in the RPR network. However, because it's difficult to directly regress to distance from an image, we use the output transformation of Eigen *et. al* [29] and use $d = \frac{1}{\sigma(\hat{d})} - 1$ where $\hat{d}$ is the regressed distance value. The distance regression layer generates *k* outputs, where *k* is the number of 2D anchor boxes used in the RPN network at each location on the feature map. We use a cross entropy loss for object classification and a Smooth L1 loss for box distance regressor layers.

C. Distance Refinement

The outputs of the radar and image proposal networks need to be merged for the second stage of the object detection network. Before using the proposals in the next stage, redundant proposals are removed by applying Non-Maximum Suppression (NMS). The NMS would normally remove overlapping proposals without discriminating based on the bounding box's origin, but we note that radar-based proposals have more reliable distance information than the image-based proposals. This is because image-based distances are estimated only from 2D image feature maps with no depth information. To make sure the radar-based distances are not unnecessarily discarded in the NMS process, we first calculate the Intersection over Union (IoU) between radar and image proposals. Next we use an IoU threshold to find the matching proposals, and overwrite the image-based distances by their radar-based counterparts for these matching proposals. The calculated IoU values are reused in the next step where NMS is applied to all proposals, regardless of their origin. The remaining proposals are then fed into the second stage of the detection network to calculate the object class and score.

D. Second Stage Detection Network

The inputs to the second stage detection network are the feature map from the image and object proposals. The structure of this network is similar to Fast R-CNN [13]. The feature map is cropped for every object proposals and is fed into the RoI pooling layer to obtain feature vectors of the same size for all proposals. These feature vectors are further processed by a set of fully connected layers and are passed to the softmax and bounding box regression layers. The output is the category classification and bounding box regression for each proposal, in addition to the distance associated to every detected object. Similar to the RPN network, we use a cross entropy loss for object classification and a Smooth L1 loss for the box regression layer.

E. Loss Function

We follow Faster R-CNN [14] and use the following multi-task loss as our objective function:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

where i is the anchor index, p_i is the i'th anchor's objectness score, p_i^* is the ground truth score (1 if anchor is positive and 0 if negative), t_i is the vector of 4 parameters representing the predicted bounding box and t_i^* is the ground truth bounding box. We use the log loss over two classes for the classification loss L_{cls}, and the the smooth L_1 loss for the regression loss, L_{reg}. N_{cls} and N_{reg} are normalization factors and λ is a balancing parameter.

IV. Experiments

A. Dataset and Implementation Details

Our network uses FPN [17] with ResNet-50 [30] pre-trained on ImageNet as the backbone for image feature extraction. We use the same RPN architecture as Faster R-CNN [14], and only add the distance regression layer on top of its convolution layer for distance estimation. For the second stage of the network, the classification stage, we use the same architecture as Fast R-CNN.

TABLE I: Performance on the nuScenes validation set.

	Weighted AP	AP	AP50	AP75	AR	MAE
Faster R-CNN	No	34.95	58.23	36.89	40.21	-
RRPN	No	35.45	59.00	37.00	**42.10**	-
Ours	No	**35.60**	**60.53**	**37.38**	**42.10**	2.65
Faster R-CNN	Yes	43.78	-	-	-	-
CRF-Net	Yes	43.95	-	-	-	-
Ours	Yes	**44.49**	-	-	-	-

TABLE II: Per-class performance

	Car	Truck	Person	Bus	Bicycle	Motorcycle
Faster R-CNN	51.46	33.26	27.06	47.73	24.27	25.93
RRPN	41.80	**44.70**	17.10	**57.20**	21.40	**30.50**
Ours	**52.31**	34.45	**27.59**	48.30	**25.00**	25.97

TABLE III: Per-class Mean Absolute Error (MAE) for distance estimation

Category	Car	Truck	Person	Bus	Bicycle	Motorcycle
MAE	2.66	3.26	2.99	3.187	1.97	2.81

We use the nuScenes dataset [9] to evaluate our network. Out of 23 different object classes in this dataset, we use 6 classes as shown in Table II. The nuScenes dataset includes data from 6 different cameras and 5 radars mounted on the vehicle. We use samples from the front- and rear-view cameras together with detection from all the radars for both training and evaluation. The ground truth annotations in the nuScenes dataset are provided in the form of 3D boxes in the global coordinate system. As a preprocessing step, we first transform the annotations and radar point clouds to the vehicle coordinate, then convert all 3D annotations to their equivalent 2D bounding boxes. This is achieved by mapping the 3D boxes to the image and finding the smallest 2D enclosing bounding box. For every 3D annotation, we also calculate the distance from vehicle to the box and use it as the ground truth distance for its 2D counterpart. The official nuScenes splits are used for training and evaluation, and images are used at their original resolution (900×1600) for both steps. No data augmentation is used as the number of labeled instances for each category is relatively large. We used PyTorch to implement our network and all experiments were conducted on a computer with two Nvidia Quadro P6000 GPUs.

B. Evaluation

The performance of our method is shown in Table I. This table shows the overall Average Precision (AP) and Average Recall (AR) for the detection task, and Mean Absolute Error for the distance estimation task. We use the Faster R-CNN network as our image-based detection baseline, and compare our results with RRPN [27] and CRF-Net[28], which use radar and camera fusion for object detection. CRF-Net only uses images from the front-view camera and also uses a weighted AP score based on the number of object appearances in the dataset. For fair comparison, we use the weighted AP scores to compare our results with this network. The CRF-Net also reports some results after filtering the ground truth to consider only objects that are detected by at least one radar, and filtering radar detections that are outside 3D ground truth bounding boxes. We do not apply these filtering operations and only compare with their results on the unfiltered data. Since CRF-Net does not report AR, per-class AP, or AP for different IoU levels, we only compare our overall AP with theirs.

According to Table I our method outperforms RRPN and CRF-Net for the detection task, improving the AP score by 0.15 and 0.54 points respectively. Our proposed method also accurately estimates the distance for all detected objects, as visualized in Fig. 4. We use Mean Absolute Error (MAE) as the evaluation metric for distance estimation. Our method achieves an MAE of 2.65 on all images. The per-class MAE values are provided in Table III. According to this table, larger objects such as trucks and buses have a higher distance error compared to other classes. This behavior is expected and could be explained by the fact that radars usually report multiple detections for larger objects, which results in several object proposals with different distances for the same object. Additionally, most radar detections happen to be at the edge of objects, while the ground truth distances are measured from the center of objects. This results in higher distance mismatch error for larger objects, where the distance between the edge and center of the object is significant.

V. Conclusion and Future Work

We proposed a radar-camera fusion algorithm for joint object detection and distance estimation for autonomous driving scenarios. The proposed architecture uses a multi-modal fusion approach to employ radar point clouds and image feature maps to generating accurate object proposals. The proposed network also uses both radar detections and image features for distance estimation for every generated proposal. These proposals are fed into the second stage of the detection network for object classification. Experiments on the nuScenes dataset show that our method outperforms other radar-camera fusion-based object detection methods, while at the same time accurately estimates the distance to every detection.

As a future work, we intend to work on reducing the distance error introduced by the mismatch between radar detections and ground truth measurements. This can be alleviated to some extent by a pre-processing step, where the ground truth distances are re-calculated based on the distance between the edge of the bounding boxes to the vehicle. Additionally, a clustering algorithm could be used to group the Radar detections and reduce the distance error introduced by having multiple detections for larger objects.

References

[1] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2017.691

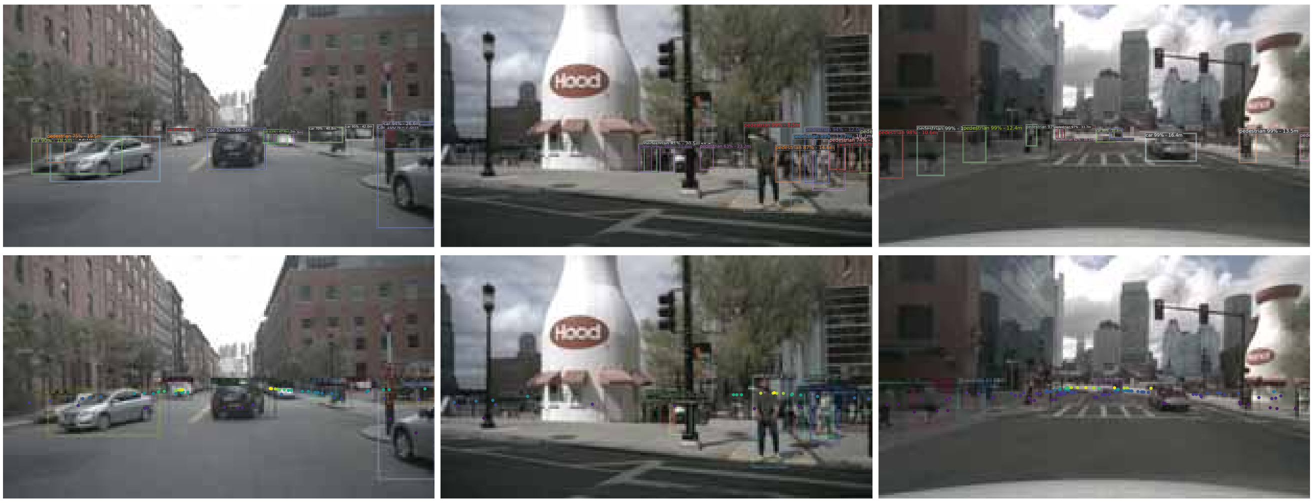


Fig. 4: Object detection and distance estimation results. Top: detection outputs, Bottom: ground truth. (Best viewed in color and zoomed-in)

[3] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018. [Online]. Available: http://dx.doi.org/10.1109/iros.2018.8594049

[4] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *arXiv preprint arXiv:1608.07916*, 2016.

[5] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 2018.

[6] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[9] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[10] D. Feng, C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, F. Duffhauss, C. Glaeser, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *arXiv preprint arXiv:1902.07830*, 2019.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[12] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[13] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[18] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1513–1518.

[19] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7276–7282.

[20] K. Werber, M. Rapp, J. Klappstein, M. Hahn, J. Dickmann, K. Dietmayer, and C. Waldschmidt, "Automotive radar gridmap representations," in *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. IEEE, 2015, pp. 1–4.

[21] T. Visentin, A. Sagainov, J. Hasch, and T. Zwick, "Classification of objects in polarimetric radar images using cnns at 77 ghz," in *2017 IEEE Asia Pacific Microwave Conference (APMC)*. IEEE, 2017, pp. 356–359.

[22] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer, "2d car detection in radar data with pointnets," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 61–66.

[23] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic segmentation on radar point clouds," in *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, pp. 2179–2186.

[24] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 244–253.

[25] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[26] Z. Ji and D. Prokhorov, "Radar-vision fusion for object classification," in *2008 11th International Conference on Information Fusion*. IEEE, 2008, pp. 1–7.

[27] R. Nabati and H. Qi, "Rrpn: Radar region proposal network for object detection in autonomous vehicles," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 3093–3097.

[28] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, "A deep learning-based radar and camera sensor fusion architecture for object detection," in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, 2019, pp. 1–7.

[29] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving

Tiago Cortinhal[1], George Tzelepis[2] and Eren Erdal Aksoy[1,2]

***Abstract*— In this paper, we introduce *SalsaNext* for the uncertainty-aware semantic segmentation of a full 3D LiDAR point cloud in real-time. *SalsaNext* is the *next* version of *SalsaNet* [1] which has an encoder-decoder architecture consisting of a set of ResNet blocks. In contrast to *SalsaNet,* we introduce a new context module, replace the ResNet encoder blocks with a new residual dilated convolution stack with gradually increasing receptive fields and add the *pixel-shuffle* layer in the decoder. Additionally, we switch from stride convolution to average pooling and also apply central dropout treatment. To directly optimize the Jaccard index, we further combine the weighted cross entropy loss with *Lovász-Softmax* loss [2]. We finally inject a Bayesian treatment to compute the *epistemic* and *aleatoric* uncertainties for each LiDAR point. We provide a thorough quantitative evaluation on the Semantic-KITTI dataset [3], which demonstrates that *SalsaNext* outperforms the previous networks and ranks first on the Semantic-KITTI leaderboard.**

I. Introduction

Scene understanding is an essential prerequisite for autonomous vehicles. Semantic segmentation helps gaining a rich understanding of the scene by predicting a meaningful class label for each individual sensory data point. Safety-critical systems, such as self-driving vehicles, however, require not only highly accurate but also reliable scene segmentation with a consistent measure of uncertainty. This is because the quantitative uncertainty measures can be propagated to the subsequent units, such as decision making modules to lead to safe manoeuvre planning or emergency braking, which is of utmost importance in safety-critical systems. Therefore, semantic segmentation predictions integrated with reliable confidence estimates can significantly reinforce the concept of safe autonomy.

In this work, we introduce a novel neural network architecture to perform uncertainty-aware semantic segmentation of a full 3D LiDAR point cloud in real-time. Our proposed network is built upon the *SalsaNet* model [1], hence, named *SalsaNext.* The base *SalsaNet* model has an encoder-decoder skeleton where the encoder unit consists of a series of ResNet blocks and the decoder part upsamples and fuses features extracted in the residual blocks. In *SalsaNext,* our contributions lie in the following aspects:

- To capture the global context information in the full 360° LiDAR scan, we introduce a new context module before encoder, which consists of a residual dilated convolution stack fusing receptive fields at various scales.
- To increase the receptive field, we replaced the ResNet block in the encoder with a novel combination of a set of dilated convolutions (with a rate of 2) each of which has different kernel sizes $(3, 5, 7)$. We concatenated the convolution outputs and combined with residual connections yielding a branch-like structure.
- To avoid any checkerboard artifacts in the upsampling process, we replaced the transposed convolution layer in the *SalsaNet* decoder with a *pixel-shuffle* layer [4] which directly leverages on the feature maps to upsample the input with less computation.
- To boost the roles of very basic features (e.g. edges and curves) in the segmentation process, the dropout treatment was altered by omitting the first and last network layers in the dropout process.
- To have a lighter model, average pooling was employed instead of having stride convolutions in the encoder.
- To enhance the segmentation accuracy by optimizing the Jaccard index, the weighted cross entropy loss was combined with the *Lovász-Softmax* loss [2].
- To further estimate the *epistemic* (model) and *aleatoric* (observation) uncertainties for each 3D LiDAR point, the deterministic *SalsaNet* model was transformed into a stochastic format by applying the Bayesian treatment.

All these contributions form the here introduced *SalsaNext* model which is the probabilistic derivation of the *SalsaNet* with a significantly better segmentation performance. The input of *SalsaNext* is the rasterized image of the full LiDAR scan in the panoramic view. The final network output is the point-wise classification scores together with uncertainty measures. To the best of our knowledge, this is the first work showing the both *epistemic* and *aleatoric* uncertainty estimation on the LiDAR point cloud segmentation task.

Quantitative and qualitative experiments on the Semantic-KITTI dataset [3] show that the proposed *SalsaNext* significantly outperforms other state-of-the-art networks in terms of pixel-wise segmentation accuracy while having much fewer parameters, thus requiring less computation time. *SalsaNext* ranks first place on the Semantic-KITTI leaderboard. We release our source code and trained model to encourage research on the subject [1].

II. Related Work

As comprehensively described in [5], there exists two mainstream deep learning approaches addressing the seman-

[1]Halmstad University, School of Information Technology, Center for Applied Intelligent Systems Research, Halmstad, Sweden

[2]Volvo Technology AB, Volvo Group Trucks Technology, Vehicle Automation, Gothenburg, Sweden

[1] https://github.com/TiagoCortinhal/SalsaNext

tic segmentation of 3D LiDAR data only: point-wise and projection-based neural networks.

Point-wise methods [6], [7] directly process the raw irregular 3D points without applying any additional transformation or pre-processing. Shared multi-layer perceptron-based PointNet [6], the subsequent work PointNet++ [7], and *superpoint* graph SPG networks [8] are considered in this group. Although such methods are powerful on small point clouds, their processing capacity and memory requirement, unfortunately, becomes inefficient when it comes to the full 360° LiDAR scans.

Projection-based methods instead transform the 3D point cloud into various formats such as voxel cells [9], [10], [11], multi-view representation [12], lattice structure [13], [14], and rasterized images [1], [15], [16], [17]. For instance, voxel-based methods discretize the 3D space into 3D volumetric space and assign each point to the corresponding voxel. Sparsity and irregularity in point clouds, however, yield redundant computations since many voxel cells may stay empty. A common attempt to overcome this sparsity problem is to project 3D point clouds into 2D image space either in the Bird-Eye-View [1], [18], [19] or spherical Range-View (RV) [20], [15], [16], [17], [21]. Unlike point-wise and other projection-based approaches, such 2D rendered image representations are more compact, dense and computationally cheaper as they can be processed by standard 2D convolutionals. Therefore, our *SalsaNext* model projects the LiDAR point cloud into 2D RV image.

Bayesian Neural Networks (BNNs) learn approximate distribution on the weights to further generate uncertainty estimates. There are two types of uncertainties: *Aleatoric* which can quantify the intrinsic uncertainty coming from the observed data, and *epistemic* where the model uncertainty is estimated by inferring with the posterior weight distribution, usually through Monte Carlo sampling. Bayesian modelling helps estimating both uncertainty types.

Gal *et al.* [22] proved that dropout can be used as a Bayesian approximation to estimate the uncertainty in classification, regression and reinforcement learning tasks while this idea was also extended to semantic segmentation of RGB images by Kendall *et al.* [23]. Loquercio *et al.* [24] proposed a framework which extends the dropout approach by propagating the uncertainty that is produced from the sensors through the activation functions without the need of retraining. Recently, both uncertainty types were applied to 3D point cloud object detection [25] and optical flow estimation [26] tasks. To the best of our knowledge, BNNs have not been employed in modeling the uncertainty of semantic segmentation of 3D point clouds, which is one of the main contributions in this work.

III. Method

SalsaNext is built upon the base *SalsaNet* model [1] which follows the standard encoder-decoder architecture with a bottleneck compression rate of 16. The original *SalsaNet* encoder contains a series of ResNet blocks each of which is followed by dropout and downsampling layers. The decoder blocks apply transpose convolutions and fuse upsampled features with that of the early residual blocks via skip connections. To further exploit descriptive spatial cues, a stack of convolution is inserted after the skip connection. We, in this study, improve the base structure of *SalsaNet* with the following contributions:

Point Cloud Representation: We project the unstructed 3D LiDAR point cloud onto a spherical surface to generate the LIDAR's native Range View (RV) image. This leads to dense and compact representation which allows standard convolution operations. Following the work of [20], we considered the full 360° field-of-view in the projection process. During the projection, 3D point coordinates (x, y, z), the intensity value (i) and the range index (r) are stored as separate RV image channels. This yields a $[w \times h \times 5]$ image.

Contextual Module: The global context information gathered by larger receptive fields plays a crucial role in learning complex correlations between classes [29]. To aggregate the context information in different regions, we place a residual dilated convolution stack that fuses a larger receptive field with a smaller one by adding 1×1 and 3×3 kernels right at the beginning of the network. This helps us capture the global context alongside with more detailed spatial information.

Dilated Convolution: Receptive fields play a crucial role in extracting spatial features. A straightforward approach to capture more descriptive spatial features would be to enlarge the kernel size. This has, however, a drawback of increasing the number of parameters drastically. Instead, we replace the ResNet blocks in the original *SalsaNet* encoder with a novel combination of a set of dilated convolutions having effective receptive fields of $3, 5$ and 7. We further concatenate each dilated convolution output and apply a 1×1 convolution followed by a residual connection in order to let the network exploit more information from the fused features coming from various depths in the receptive field. Each of these new residual dilated convolution blocks is followed by dropout and pooling layers.

Pixel-Shuffle Layer: The original *SalsaNet* decoder involves transpose convolutions which are computationally expensive layers in terms of number of parameters. We replace these standard transpose convolutions with the *pixel-shuffle* layer [4] which leverages on the learnt feature maps to produce the upsampled feature maps by shuffling the pixels from the channel dimension to the spatial dimension. More precisely, the *pixel-shuffle* operator reshapes the elements of $(H \times W \times Cr^2)$ feature map to a form of $(Hr \times Wr \times C)$, where $H, W, C,$ and r represent the height, width, channel number and upscaling ratio, respectively. We additionally double the filters in the decoder side and concatenate the *pixel-shuffle* outputs with the skip connection before feeding them to the additional dilated convolutional blocks.

Central Encoder-Decoder Dropout: Lower network layers extract basic features such as edges and corners which are consistent over the data distribution and dropping out these layers will prevent the network to properly form the higher level features in the deeper layers. We, therefore, insert dropout only to the central encoder and decoder layers

	Approach	Size	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	**mean-IoU**
Point-wise	Pointnet [6]	50K pts	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.6
	Pointnet++ [7]		53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1
	TangentConv [27]		86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2	35.9
	RandLa-Net [28]		**94.2**	26.0	25.8	**40.1**	**38.9**	49.2	48.2	7.2	90.7	60.3	73.7	**38.9**	86.9	56.3	81.4	61.3	**66.8**	49.2	47.7	53.9
	LatticeNet [14]		92.9	16.6	22.2	26.6	21.4	35.6	43.0	**46.0**	90.0	59.4	74.1	22.0	88.2	58.8	81.7	**63.6**	63.1	51.9	48.4	52.9
Projection-based	SqueezeSeg [15]	64×2048 pixels	68.8	16.0	4.1	3.3	3.6	12.9	13.1	0.9	85.4	26.9	54.3	4.5	57.4	29.0	60.0	24.3	53.7	17.5	24.5	29.5
	SqueezeSegV2 [16]		81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	39.7
	RangeNet53++ [20]		91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	**91.8**	**65.0**	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9	52.2
	3D-MiniNet [21]		90.5	42.3	**42.1**	28.5	29.4	47.8	44.1	14.5	91.6	64.2	74.5	25.4	89.4	60.8	**82.8**	60.8	66.7	48.0	56.6	55.8
	SqueezeSegV3 [17]		92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9	55.9
	SalsaNet [1]	64×2048 pixels	87.5	26.2	24.6	24.0	17.5	33.2	31.1	8.4	89.7	51.7	70.7	19.7	82.8	48.0	73.0	40.0	61.7	31.3	41.9	45.4
	SalsaNext [Ours]		91.9	**48.3**	38.6	38.9	31.9	**60.2**	**59.0**	19.4	91.7	63.7	**75.8**	29.1	**90.2**	**64.2**	81.8	**63.6**	66.5	**54.3**	**62.1**	**59.5**

TABLE I

Quantitative comparison on Semantic-KITTI test set (sequences 11 to 21). IoU scores are given in percentage (%).

which leads to higher network performance.

Average Pooling: In the base *SalsaNet* model the down-sampling was performed via a strided convolution which introduces additional learning parameters. Given that the down-sampling process is relatively straightforward, we hypothesize that learning at this level would not be needed. Thus, to allocate less memory *SalsaNext* switches to average pooling for the downsampling.

Uncertainty Estimation: In *SalsaNext,* the *epistemic* uncertainty is computed using the weight's posterior which is approximated by using dropout as shown in [22]. By following the work in [24], we compute the optimal dropout rate for an *already trained network* by applying a grid search on a log-range of a certain number of possible rates. To measure the *epistemic* uncertainty, we employ a Monte Carlo sampling during inference: we run n trials with this optimal dropout rate and compute the average of the variance of the n predicted outputs. To be able to track the *aleatoric* uncertainty, we propagate the known LiDAR noise characteristic through the network via Assumed Density Filtering (ADF) [30]. A forward pass in this ADF-based modified network finally generates output predictions with their respective aleatoric uncertainties [24].

Loss: To cope with the imbalanced class problem, we follow the same strategy in *SalsaNet* and add more value to the under-represented classes by weighting the softmax cross-entropy loss with the inverse square root of class frequency. This reinforces the network response to the classes appearing less in the dataset. In contrast to *SalsaNet,* we here also incorporate the *Lovász-Softmax* loss [2] in the learning procedure to maximize the intersection-over-union (IoU) score, i.e. the Jaccard index. The IoU metric is the most commonly used metric to evaluate the segmentation performance. Nevertheless, IoU is a discrete and not derivable metric that does not have a direct way to be employed as a loss. In [2], the authors adopt this metric with the help of the Lovász extension for submodular functions. Finally, the total loss function of *SalsaNext* is a linear combination of weighted cross-entropy and *Lovász-Softmax* losses.

Optimizer and Regularization: As an optimizer, we employed stochastic gradient descent with an initial learning rate of 0.01 which is decayed by 0.01 after each epoch. We also applied an L2 penalty with $\lambda = 0.0001$ and a momentum of 0.9. The batch size and spatial dropout probability were fixed at 24 and 0.2, respectively. To prevent overfitting, we augmented the data by applying a random rotation/translation, flipping randomly around the y-axis and randomly dropping points before creating the projection. Every augmentation is applied independently of each other with a probability of 0.5.

Post-processing: We further applied the kNN-based post-processing technique [20] to prevent the projection-based information loss when the RV image is re-projected back to the original 3D space.

IV. Experiments

We evaluate the performance of *SalsaNext* and compare with the state-of-the-art semantic segmentation methods on the large-scale challenging Semantic-KITTI dataset [3] which provides over 43K LiDAR data. Obtained quantitative results compared to state-of-the-art point-wise and projection-based approaches are reported in Table I. Our *SalsaNext* model considerably outperforms the others by leading to the highest mean IoU score (59.5%) which is $+3.6\%$ over the previous state-of-the-art method [17]. In contrast to the original *SalsaNet,* we obtain 14% improvement.

Following the work of [24], we further computed the *epistemic* and *aleatoric* uncertainty without retraining the *SalsaNext* model. Fig. 1 depicts the quantitative relationship between the *epistemic* (model) uncertainty and the number of points that each class has in the Semantic-KITTI test set. This plot has diagonally distributed samples, which clearly shows

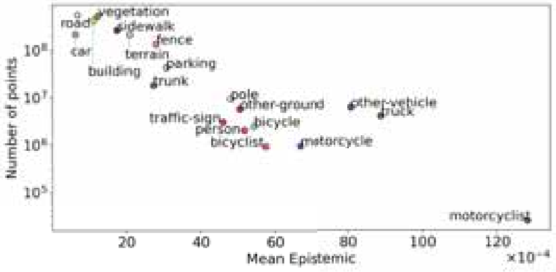


Fig. 1. The relationship between the *epistemic* uncertainty and the number of points (in log scale) in each class.

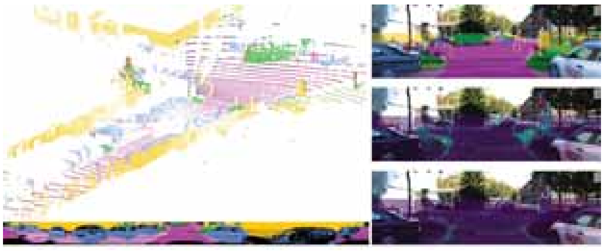


Fig. 2. A sample qualitative result. At the bottom, the range-view image of the network response is shown. The top camera image on the right shows the projected segments whereas the middle and bottom images depict the projected *epistemic* and *aleatoric* uncertainties, respectively. Note that the lighter the color is, the more uncertain the network becomes.

that the network becomes less certain about rare classes represented by low number of points (e.g. motorcyclist).

Fig. 2 shows sample qualitative segmentation and uncertainty results. In this figure, only for visualization purposes, segmented object points are also projected back to the respective camera image. Note that these camera images have not been used for training of *SalsaNext.* As depicted in Fig. 2, *SalsaNext* can, to a great extent, distinguish road, car, and other object points. In Fig. 2, we additionally show the estimated *epistemic* and *aleatoric* uncertainty values projected on the camera image for the sake of clarity. In line with Fig. 1, we obtain high *epistemic* uncertainty for rare classes such as other-ground (see Fig. 2). We also observe that high level of *aleatoric* uncertainty mainly appears around segment boundaries and on distant objects as shown in Fig. 2. In the supplementary video[2], we provide more qualitative results.

V. Conclusion

We presented a new uncertainty-aware semantic segmentation network that can process the full 360° LiDAR scan in real-time. *SalsaNext* builds up on *SalsaNet* and can achieve over 14% more accuracy. In contrast to state-of-the-art methods, *SalsaNext* returns $+3.6\%$ better mIoU score. *SalsaNext* can also estimate both data and model-based uncertainty.

References

[1] E. E. Aksoy, S. Baci, and S. Cavdar, “Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving,” in *IEEE Intelligent Vehicles Symposium (IV2020)*, 2020.

[2] M. Berman, A. Rannen Triki, and M. B. Blaschko, “The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks,” in *CVPR*, 2018.

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences,” in *ICCV*, 2019.

[4] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” *CoRR*, vol. abs/1609.05158, 2016.

[5] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, “Deep learning for 3d point clouds: A survey,” *CoRR*, 2019.

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *CVPR*, 2017.

[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *NIPS*, 2017.

[8] L. Landrieu and M. Simonovsky, “Large-scale point cloud semantic segmentation with superpoint graphs,” in *CVPR*, 2018.

[9] C. Zhang, W. Luo, and R. Urtasun, “Efficient convolutions for real-time semantic segmentation of 3d point clouds,” in *3DV*, 2018.

[10] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in *CVPR*, 2018.

[11] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese, “Segcloud: Semantic segmentation of 3d point clouds,” in *3DV*, 2017.

[12] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg, “Deep projective 3d semantic segmentation,” *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1705.03428

[13] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M. Yang, and J. Kautz, “Splatnet: Sparse lattice networks for point cloud processing,” in *CVPR*, 2018.

[14] R. Alexandru Rosu, P. Schütt, J. Quenzel, and S. Behnke, “LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices,” *arXiv e-prints*, p. arXiv:1912.05905, Dec. 2019.

[15] B. Wu, A. Wan, X. Yue, and K. Keutzer, “Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud,” *ICRA*, 2018.

[16] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, “Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud,” in *ICRA*, 2019.

[17] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, “Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation,” 2020.

[18] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, “Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving,” *IEEE RAL*, vol. 3, no. 4, pp. 3434–3440, Oct 2018.

[19] M. Simon, S. Milz, K. Amende, and H. Gross, “Complex-yolo: Real-time 3d object detection on point clouds,” *CoRR*, 2018.

[20] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, “RangeNet++: Fast and Accurate LiDAR Semantic Segmentation,” in *IROS*, 2019.

[21] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, “3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation,” 2020.

[22] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *ICML*, 2016.

[23] A. Kendall, V. Badrinarayanan, and R. Cipolla, “Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding,” *arXiv preprint arXiv:1511.02680*, 2015.

[24] A. Loquercio, M. Segú, and D. Scaramuzza, “A general framework for uncertainty estimation in deep learning,” *RA-L*, 2020.

[25] D. Feng, L. Rosenbaum, and K. Dietmayer, “Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection,” in *ITSC*. IEEE, 2018, pp. 3266–3273.

[26] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, “Uncertainty estimates and multi-hypotheses networks for optical flow,” in *ECCV*, 2018, pp. 652–667.

[27] M. Tatarchenko, J. Park, V. Koltun, and Q. Zhou, “Tangent convolutions for dense prediction in 3d,” in *CVPR*, 2018.

[28] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, “Randla-net: Efficient semantic segmentation of large-scale point clouds,” 2019.

[29] R. P. K. Poudel, S. Liwicki, and R. Cipolla, “Fast-scnn: Fast semantic segmentation network,” *CoRR*, vol. abs/1902.04502, 2019.

[30] J. Gast and S. Roth, “Lightweight probabilistic deep networks,” in *CVPR*, 2018, pp. 3369–3378.

[31] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *CVPR*, 2012.

[2] https://youtu.be/MlSaIcD9ItU

VI. Supplementary Material

A. Ablation Study

In this ablative analysis, we investigate the individual contribution of each improvements over the original *SalsaNet* model. Table II shows the total number of model parameters and FLOPs (Floating Point Operations) with the obtained mIoU scores on the Semantic-KITTI test set before and after applying the kNN-based post processing.

As depicted in Table II, each of our contributions on *SalsaNet* has a unique improvement in the accuracy. The post processing step leads to a certain jump (around 2%) in the accuracy. The peak in the model parameters is observed when dilated convolution stack is introduced in the encoder, which is vastly reduced after adding the *pixel-shuffle* layers in the decoder. Combining the weighted cross-entropy loss with *Lovász-Softmax* leads to the highest increment in the accuracy as the Jaccard index is directly optimized. We can achieve the highest accuracy score of 59.5% by having only 2.2% (i.e. 0.15M) extra parameters compared to the original *SalsaNet* model. Table II also shows that the number of FLOPs is correlated with the number of parameters. We note that adding the *epistemic* and *aleatoric* uncertainty computations do not introduce any additional training parameter since they are computed after the network is trained.

B. Runtime Evaluation

Runtime performance is of utmost importance in autonomous driving. Table III reports the total runtime performance for the CNN backbone network and post-processing module of *SalsaNext* in contrast to other networks. To obtain fair statistics, all measurements are performed using the entire Semantic-KITTI dataset on the same single NVIDIA Quadro RTX 6000 - 24GB card. As depicted in Table III, our method clearly exhibits better performance compared to, for instance, RangeNet++ [20] while having 7× less parameters. *SalsaNext* can run at 24 Hz when the uncertainty computation is excluded for a fair comparison with deterministic models. Note that this high speed we reach is significantly faster than the sampling rate of mainstream LiDAR sensors which typically work at 10 Hz [31]. Fig. 3 also compares the overall performance of *SalsaNext* with the other state-of-the-art semantic segmentation networks in terms of runtime, accuracy, and memory consumption.

As illustrated in Fig. 3, there is a clear split between projection-based and point-wise networks in terms of accuracy, runtime and memory consumption. For instance, projection-based approaches (shown in green circles in Fig. 3) achieve the state-of-the-art accuracy while running significantly faster. Although point-wise networks (red squares) have slightly lower number of parameters, they cannot efficiently scale up to large point sets due to the limited processing capacity, thus, they take a longer runtime. *SalsaNext* falls into the projection-based networks and achieves the highest score while achieving real-time performance with relatively low number of parameters. It is also highly important to note that unlike *SalsaNext,* both point-wise and projection-based approaches in Fig. 3 lack uncertainty measures, i.e. confidence scores, for their predictions.

	mean IoU (w/o kNN)	mean IoU (+kNN)	Number of Parameters	FLOPs
SalsaNet [1]	43.5	44.8	6.58 M	51.60 G
+ context module	44.7	46.0	6.64 M	69.20 G
+ central dropout	44.6	46.3	6.64 M	69.20 G
+ average pooling	47.7	49.9	5.85 M	66.78 G
+ dilated convolution	48.2	50.4	9.25 M	161.60 G
+ Pixel-Shuffle	50.4	53.0	6.73 M	125.68 G
+ *Lovász-Softmax* loss	**56.6**	**59.5**	6.73 M	125.68 G

TABLE II

Ablative analysis.

	Processing Time (msec)					
	CNN	kNN	Total	Speed (fps)	Parameters	FLOPs
RangeNet++ [20]	63.51	2.89	66.41	15 Hz	50 M	720.96 G
SalsaNet [1]	35.78	2.62	38.40	26 Hz	6.58 M	51.60 G
SalsaNext [Ours]	38.61	2.65	41.26	24 Hz	6.73 M	125.68 G

TABLE III

Runtime performance on the Semantic-KITTI test set

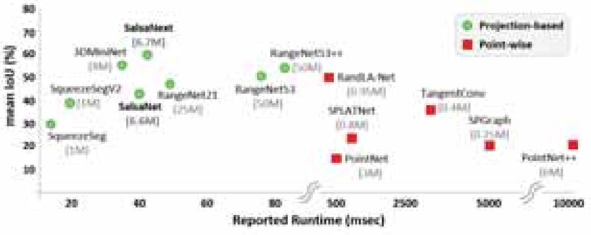
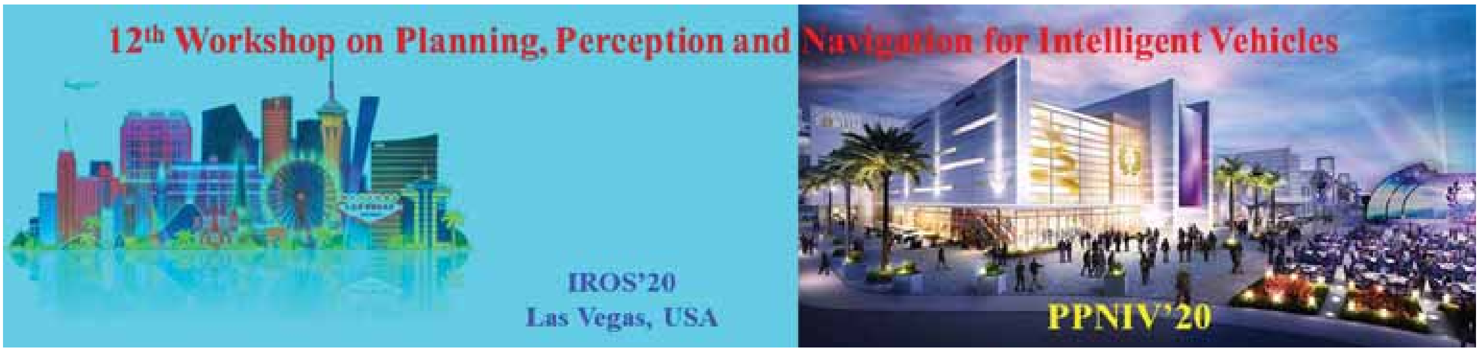


Fig. 3. Mean IoU versus runtime plot for the state-of-the-art 3D point cloud semantic segmentation networks on the Semantic-KITTI dataset [3]. Inside parentheses are given the total number of network parameters in Millions. All deep networks visualized here use only 3D LiDAR point cloud data as input. Note that only the published methods are considered.


12th Workshop on Planning, Perception and Navigation for Intelligent Vehicles
IROS'20
Las Vegas, USA
PPNIV'20

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems

SDVTracker: Real-Time Multi-Sensor Association and Tracking for Self-Driving Vehicles

Shivam Gautam[1], Gregory P. Meyer[1], Carlos Vallespi-Gonzalez[1] and Brian C. Becker[1]

***Abstract*— Accurate motion state estimation of Vulnerable Road Users (VRUs), is a critical requirement for autonomous vehicles that navigate in urban environments. Due to their computational efficiency, many traditional autonomy systems perform multi-object tracking using Kalman Filters which frequently rely on hand-engineered association. However, such methods fail to generalize to crowded scenes and multi-sensor modalities, often resulting in poor state estimates which cascade to inaccurate predictions. We present a practical and lightweight tracking system, SDVTracker, that uses a deep learned model for association and state estimation in conjunction with an Interacting Multiple Model (IMM) filter. The proposed tracking method is fast, robust and generalizes across multiple sensor modalities and different VRU classes. In this paper, we detail a model that jointly optimizes both association and state estimation with a novel loss, an algorithm for determining ground-truth supervision, and a training procedure. We show this system significantly outperforms hand-engineered methods on a real-world urban driving dataset while running in less than 2.5 ms on CPU for a scene with 100 actors, making it suitable for self-driving applications where low latency and high accuracy is critical.**

I. Introduction

Self-Driving Vehicles (SDVs) depend on a robust autonomy system to perceive actors and anticipate future actions in order to accurately navigate the world. Interacting well with Vulnerable Road Users (VRUs) [1] such as pedestrians and bicyclists requires good motion estimates. A classical autonomy system that uses structured prediction for actor trajectory prediction [2], [3], [4] needs not only high detection rates to identify objects in the scene, but also robust tracking performance to estimate the motion state. Probabilistic tracking using filters can be a reliable method to estimate the motion state [5]. These methods attempt to refine the motion estimates of previously tracked objects by associating them with a given set of detections in the scene at the current timestamp.

Failures in association cause inaccurate state estimates, often leading to cascading errors in future associations, state estimations, and trajectory predictions resulting in improper autonomy behavior [6]. In simple scenes, engineered solutions do well. However, associating VRUs in crowded, urban environments is challenging due to occlusions, crowd density, varying motions and intermittent detector false positives or false negatives. Any errors in association break the strict assumption for probabilistic filtering regarding observations belonging to the same actor, leading to egregious errors. Incorporating detectors for additional sensor modalities, such as LiDAR and camera detectors, improves overall recall, but increases the likelihood of mis-association, especially as each sensor has different failure modes and noise characteristics. Learned approaches offer improved performance, but are often restricted to the 2D image plane [7], require a fixed number of objects [8], need expensive feature extraction on specialized GPU hardware [9], or can run only offline [10], making them unsuitable to self-driving applications.

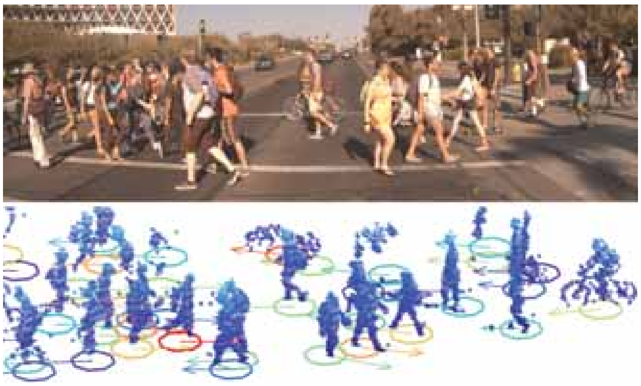


Fig. 1. Association and tracking of pedestrians is challenging in dense, urban environments. We propose a real-time learned association and tracking system with IMM filtering that incorporates LiDAR + camera modalities and show improvements on the task of both association and state estimation.

To address these limitations, we propose SDVTracker, a learned association and tracking system for improving motion estimation of VRUs in challenging, self-driving domains. Fig. 1 demonstrates our approach performing well in dense crowds across many classes of VRUs including pedestrians, bicyclists, and skateboarders. As the number of VRUs in the scene increases, we show that this method scales better than classical approaches. Our approach generalizes to multi-sensor tracking, improving recall and tracking when both LiDAR and camera detections are used as asynchronous input. In addition to learning association, we propose a novel method to jointly estimate association and state, which leads to improved performance. Further, we show a method of incorporating our learned association and state within a tracking system that uses an Interacting Multiple Model (IMM) filter. Finally, SDVTracker offers real-time performance on commodity CPUs, making it well-suited for compute-limited platforms.

This work is part of the work done at Uber Advanced Technologies Group.

[1] All authors work with the Perception team at Uber ATG. {sgautam, gmeyer, cvallespi, bbecker}@uber.com

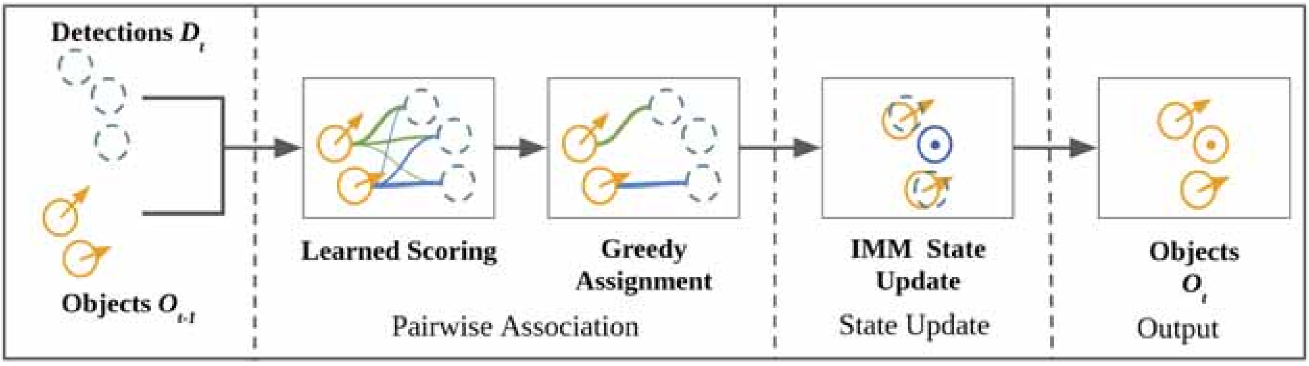


Fig. 2. Overview of the association and tracking system. SDVTracker scores the candidate pairs with a learned model to estimate the association probability. After enforcing 1-to-1 correspondence through greedy assignment, we use the learned associations and motion estimates as observations within an IMM update.

II. Related Work

As more autonomous capabilities are added to vehicles, it is critical for these intelligent vehicles to understand and predict the behavior of humans that they interact with to operate safely. Ohn-Bar and Trivedi [11] provide a thorough survey into three areas of active research where humans and automated vehicles interact, including humans inside the intelligent vehicle, humans around the vehicle, and humans operating surrounding vehicles. In this work, we focus on understanding the motion of humans around the SDV.

A. Filter-based Tracking

A conventional algorithm to perform the motion state estimation from observations is the Kalman Filter (KF) [12]. This algorithm works in two steps that get applied recursively: *prediction* and *update*. In the *prediction* step, the filter produces estimates of the state variables and their uncertainties. The *update* step is performed when the new measurement arrives, in which the filter corrects the state by combining the new measurement and the filter prediction weighted by their certainties. This filter, and its variants, are a common class of filter-based methods [13], and are widely used due to their ability to produce better state estimates than those based on a single measurement. However, the KF is limited to linear functions for the state transition as well as the observation model. In our case, this reduces our ability to correctly track objects that can have non-linear motions, such as accelerations. The Extended Kalman Filter (EKF) overcomes this constraint by linearizing these functions, but it is often difficult to tune a single filter for all the motion modalities we encounter for each object. In this paper, we use the Interacting Multiple Model (IMM) [14] algorithm because it overcomes these limitations by tracking with multiple models concurrently and fusing their predictions weighted by their confidences. Furthermore, the IMM has been shown to offer performance similar to the best motion model.

B. Tracking-by-Detection

Most recent work on Multi-Object Tracking (MOT) utilize the tracking-by-detection paradigm [10], [15], [16], [17], [18], [19], [20], [7], [21], [8], [22] where detections are provided each time-step by a detector, and tracking is performed by linking detections across time. As a result, the task of object tracking becomes a data association problem. Most tracking-by-detection methods solve the association problem in one of two ways, either in an online (step-wise) fashion [15], [16], [17], [18], [20], [7], [8], [22] or in an offline (batch-wise) manner [10], [19], [21]. Online methods associate new detections at each time-step to the existing tracks, and the association is posed as a bipartite graph matching problem. On the other hand, offline methods often consider the entire sequence, and data association is cast as a network flow problem. Online methods are appropriate for real-time applications like autonomous driving where offline approaches are well-suited for offline tasks like video surveillance. In this work, we leverage a step-wise approach as we are interested in real-time autonomous navigation where computation efficiency is as important as accuracy.

C. Classical Association Techniques

To solve the data association problem, incoming detections at the current timestamp need to be paired to existing objects from the last timestamp. To avoid matching in the entire measurement space, every detection that lies within a certain region, or *gating region*, of an object is considered a candidate. A problem arises when multiple candidates fall within this region. A common way to solve this involves ranking each object-detection pair and then performing a bijective mapping. The bijective mapping forces each object to associate with only one detection. This mapping can be performed using common matching algorithms such as greedy best-first matching or the Munkre's algorithm [23].

A common method for ranking each object-detection is to score each detection based on the proximity to the predicted object [24]. Based on this, we consider three functions:

1) *Intersection-over-Union (IoU) score:* Many trackers use ranking functions based a measurement of overlap between predictions and detections [25]. This score is defined as the ratio between the area of intersection and the area of union of the detection and predicted polygons.

2) L_2 *Distance:* As part of the *update* step, the IMM needs to compute the *residual* or *innovation*, which is the difference between the predicted detection and the new detection. The L_2 norm of the *residual* can be used as a matching score.
3) *Mahalanobis Distance:* The IMM computes the *gain* or *blending* factor that determines the relative weight of the new detection in the *update* step. This *gain* is used to scale the *residual* vector, and the L_2 norm of the resulting vector can be used as a matching score. This association metric has been previously explored in [26], where it is used to filter infeasible associations.

D. Learned Association Techniques

More recently Recurrent Neural Networks (RNNs) have been used for association [8], [20], [7], which motivates our use of a RNN for association in this work. However, our proposed method and the previous work utilize RNNs in different ways. [8] uses a single Long-Short Term Memory (LSTM) to associate all detections to all tracks. However, it requires the number of objects to be fixed and known beforehand, which is not feasible for autonomous driving in urban environments. [20] uses an LSTM to estimate the affinity matrix between all detections and tracks one row at a time. Most similar to our approach is the work of Sadeghian et al. [7], who use three separate LSTMs to model the appearance, motion, and interaction of the tracked objects over time. Each track has its own memory for each of the LSTMs, and appearance, motion, and interaction features are extracted for each detection using a set of Convolution Neural Networks (CNNs). The output of the LSTMs and the CNNs are fed into a multi-layer neural network to estimate the likelihood that the detection should be associated to the track. Unlike [7], our proposed method uses a single LSTM to model multimodal features of an object over time. Furthermore, in addition to an association probability, our approach predicts a score for each possible match in order to improve association in heavily crowded scenes, and we estimate the state of the object to improve tracking. Finally, our method tracks objects in 3D where [7] tracks objects in the 2D image plane.

E. 3D Object Tracking

The vast majority of the previous work performs object tracking in the image plane [10], [15], [16], [17], [18], [19], [20], [7], [21]. However, to autonomously navigate a vehicle through the world, we need to reason about the environment in 3D or from a bird's eye view. Furthermore, the bird's eye view is a natural representation for fusing multiple sensor modalities like LiDAR, camera and RADAR. Rangesh et al. [22] extends [18] to the bird's eye view to track vehicles. In [22], vehicles are detected with an image-based detector and localized in the bird's eye view using a flat ground assumption or with 3D measurements from LiDAR when available. The life-cycle of tracks is handled through a Markov Decision Process (MDP) where the policy is learned, and tracks are associated with detections using a Support Vector Machine (SVM). In this work, our proposed method is capable of fusing detections from various sensing modalities including LiDAR and image-based detectors. Furthermore, we associate objects across sensors and time using a RNN.

III. Proposed Method

The overall architecture of the system is depicted in Fig. 2. We use detections generated at each time step independently from LiDAR and camera sensors. To generate detections from LiDAR, we use LaserNet [27], and the detections from the camera sensors are generated using RetinaNet [28]. Similar to [22] and [29], the image-based detections are then augmented with a range estimate by projecting the LiDAR points in the image plane and using the median range value of the points associated to create 3D bounding boxes.

The proposed method is depicted in Fig. 3. During inference, the model takes an object-detection pair as input, and produces its association and state. For each object, we generate a set of potential association candidates with a corresponding score. The set of potential association candidates is created by predicting an association/mis-association probability for every pair. If the probability of association is higher than mis-association, then we add the pair to our set of potential association candidates. Afterwards, we perform greedy assignment based on the predicted score to create unique object-detection associations. We refine the detections with our predicted state estimate before using them as observations in the IMM.

After updating the state for objects, we need to prune our existing hypothesis set of objects that are currently alive in the scene. Objects that have not been observed for more than τ time-steps are removed from the scene. For objects that have not been observed for $\leq \tau$ time-steps, we extrapolate their position to the next timestamp based on their past velocity.

In the following sections, we describe in detail feature extraction from detection-object pairs, the network architecture, the multi-task loss function and the ground-truth association used during training.

A. Feature Extraction

We extract three different types of features: shape, motion and difference features. The shape features include polygon length, width, height and center coordinates. The motion features include the object's previous and predicted state. The difference features, as the name suggests, are obtained by subtracting two attributes (difference in predicted object position and the detection position, difference between the object box dimensions and the detection box dimensions). We also use the timestamp and detector confidence as input to the model. While we could use a separate network for feature extraction or use the features from the internal activation layers of the detectors, we decided to utilize these lightweight features in order to keep our method real-time and sensor-agnostic.

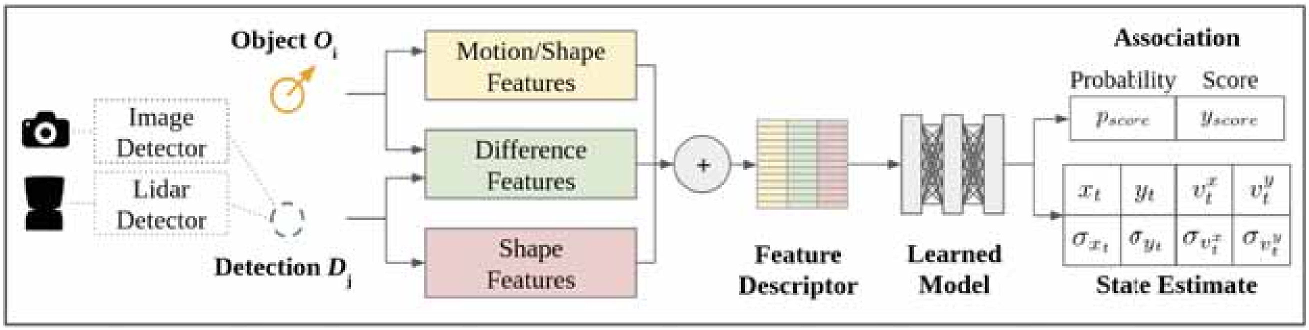


Fig. 3. Proposed architecture of the learned association and state estimation model. We perform feature extraction for each candidate pair and learn whether the pair is a true association and the posterior state estimate of the object with uncertainties. We learn a probability of association and an association score to break ties between multiple competing candidates. We show that learning both a probability and a score are beneficial to the task of association, as well as the learned posterior state estimate improves overall tracking performance.

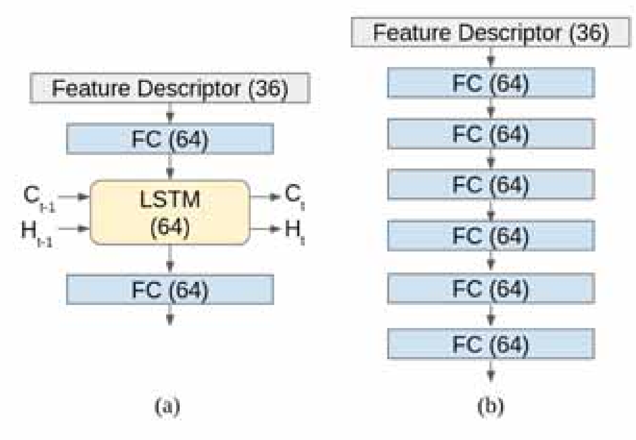


Fig. 4. Network architecture used for LSTM and MLP networks. (a) For the LSTM, we use a single LSTM cell with 64 hidden units and a single layer fully-connected encoder-decoder. The network takes the feature descriptor, the cell state (C_{t-1}) and the hidden state (H_{t-1}) for the object as input to produce association outputs and new cell (C_t) and hidden states (H_t). (b) For the MLP, we use six fully connected layers with 64 units each.

B. Learning Joint Association and Tracking

The learned model produces association probabilities, scores and state estimates. For this work, we implement a single-cell LSTM as well as a Multi-Layer Perceptron (MLP). Both network architectures can be seen in Fig. 4 and we compare the performance of each in Section IV-D.

To learn association and tracking jointly, we utilize a multi-task loss. For the task of association, we propose learning a unique training target comprised of an association probability and score. The association probability is framed as a binary classification problem in which we try to categorize candidates as associations or mis-association. The association probability is used to identify a list of potential candidates that could potentially be associated. Furthermore, the score is used to rank associations, when there are more than one potential candidates for association. The loss function for the association task is defined as,

$$\ell_{assoc} = \ell_{prob} + w_{score} \cdot \ell_{score}, \tag{1}$$

where ℓ_{prob} is the binary cross entropy used to learn the association probability, ℓ_{score} a L_2 loss on the regressed score, and w_{score} is used to weight the two losses.

In addition to learning association, we learn a posterior state update for the object. The state of the object at time t is defined as follows:

$$\boldsymbol{s}_t = [x_t, y_t, v_t^x, v_t^y] \tag{2}$$

$$\boldsymbol{\sigma}_t = [\sigma_{x_t}, \sigma_{y_t}, \sigma_{v_t^x}, \sigma_{v_t^y}] \tag{3}$$

where (x_t, y_t) is the position of the object, (v_t^x, v_t^y) is the velocity of the object, and $(\sigma_{x_t}, \sigma_{y_t}, \sigma_{v_t^x}, \sigma_{v_t^y})$ are the corresponding standard deviations. We learn the state using the following loss [30]:

$$\ell_{state} = \sum_i \left(\frac{\left(s_{t,i} - s_{t,i}^*\right)^2}{2\sigma_{t,i}^2} + \log \sigma_{t,i} \right) \tag{4}$$

where $s_{t,i}$ is the i-th element of the state vector at time t, $\sigma_{t,i}$ is the corresponding standard deviation, and $s_{t,i}^*$ is the ground-truth state. The total multi-task loss is

$$\ell_{total} = \ell_{assoc} + w_{state} \cdot \ell_{state}, \tag{5}$$

where w_{state} is used to weight the relative importance of the two tasks.

C. Training Procedure

For training the network for association and tracking, we use a dataset with time-consistent IDs for labels. To provide direct supervision for the association task, we require a function that maps a candidate object-detection pair to a binary value indicating a true or false association, along with a score.

Given a set of detections $\mathcal{D}_t = \{D_t^1, D_t^2, \ldots, D_t^N\}$ at time t and a set of objects $\mathcal{O}_{t-1} = \{O_{t-1}^1, O_{t-1}^2, \ldots, O_{t-1}^M\}$ from time $t-1$, the goal of ground-truth association is to define a mapping $f : \mathcal{O}_{t-1} \mapsto \mathcal{D}_t$ using the labeled data $\mathcal{L}_{t-1}$ and $\mathcal{L}_t$ at time $t-1$ and t. To handle the case where the object does not match to any detection, a null detection is added to $\mathcal{D}_t$. For each object $O_{t-1}^i \in \mathcal{O}_{t-1}$, we first identify the label $L_{t-1}^j \in \mathcal{L}_{t-1}$ with the maximum IoU overlap with the object. Afterwards, we find *all* detections in $\mathcal{D}_t$ with an IoU ≥ 0.1 with the label L_t^j at time t. All candidate detections are added to the training set as a true association, and their score is defined as

$$y_{score} = \|\phi(L_{t-1}^j) - \phi(O_{t-1}^i)\|_2 + \|\phi(L_t^j) - \phi(D_t^k)\|_2 \quad (6)$$

where D_t^k is a candidate detection and $\phi(\cdot)$ computes the object's centroid.

During inference the model will encounter mis-associations as well. Therefore, the model needs to learn to identify false associations. To accomplish this, we augment the dataset with examples of mis-associations. For every true association, D_t^k and O_{t-1}^i, we identify all $D_t^n \in \mathcal{D}_t$ where $\|\phi(O_{t-1}^i) - \phi(D_t^n)\|_2 < r$ and do not have an IoU ≥ 0.1 with L_t^j. We add a random subset of such examples to our dataset as false associations.

By predicting an association probability and a score, our method is robust to false positives due to duplicate detections. The probability allows us to identify all potential association candidates, including the true detection as well as false positives. The score then allows us to select the best candidate and discard the duplicate detections. In our experiments, we demonstrate the importance of predicting both.

Another advantage of breaking the problem of association into learning a probability and a score is that it eliminates the need for any engineered threshold to identify matches. Finding such thresholds can be challenging in the context of using different sources for detections with different error characteristics, e.g. image-based detections may have a higher range of uncertainty as compared to LiDAR detections. Besides, different VRU classes have different motion characteristics, e.g. bikes can move faster than pedestrians; therefore, different classes could have different scores. Our proposed method, considers all candidates with an association probability greater than the mis-association probability, and it identifies the best match with the score. As a result, we eliminate the need for any engineered thresholds.

IV. Experimental Results

A. Experimental Setup

We evaluate our method on the ATG4D dataset which contains 5,000 sequences in the training set, 900 sequences for the test set and 500 sequences for validation set. Each sequence is captured at 10Hz intervals. The data is collected using a Velodyne 64E LiDAR along with a camera sensor, while driving in an urban setting. For the experiments in the paper, we generate detections as described in Section III. To reduce the detection-object pairs that we run inference for, we prune the list of all possible pairings based on a gating radius, r. This is common practice within tracking [24] and makes the problem tractable by not considering impossible associations.

For our experiments, we set $r = 4$ m since it accommodates both slow moving pedestrians and fast moving bikes and $\tau = 5$ for our object track life management. We set $w_{score} = 0.02$ and $w_{state} = 0.06$ while training models. Finally, the individual motion models in the IMM are designed to be adapted to the different motion modalities we encounter: *static*, *constant velocity*, and *accelerating*.

B. Evaluation Metrics

We evaluate the performance of methods using standard multi-object tracking metrics [31], [32] to compare tracking methods. These include evaluating the Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP), Mostly Tracked (MT), Mostly Lost (ML) and ID Switches (IDSW). However, these metrics fail to capture the quality of velocity estimates. Measuring the accuracy of the estimated velocity is imperative to evaluating tracking performance for trackers that are used by dependent systems to predict behavior. To resolve this gap in the metrics, we propose two new metrics: Multi-Object Tracking Velocity Error (MOTVE) and Multi-Object Tracking Velocity Outliers (MOTVO).

We define MOTVE as the average velocity error for all true positive objects. This is computed as

$$\text{MOTVE} = \frac{\sum_{t=0}^{T} \sum_{i=1}^{M} ||v_t^i - \hat{v}_t^i||_2}{\sum_{t=0}^{T} g_t} \quad (7)$$

where $\hat{v}_t^i$ and v_t^i refer to the estimated velocity of i-th object and its corresponding ground-truth label at time t respectively. The number of object-label pairs present at time t are denoted by g_t.

We define MOTVO as the fraction of the object-label pairs where the velocity error is greater than a threshold,

$$\text{MOTVO} = \frac{\sum_{t=0}^{T} \sum_{i=1}^{n} \mathbf{1}[\|v_t^i - \hat{v}_t^i\|_2 > \nu]}{\sum_{t=0}^{T} g_t} \quad (8)$$

where $\mathbf{1}[\cdot]$ is an indicator function. For this evaluation, we set ν to 1 m/s for pedestrians and 1.5 m/s for bicyclists. This measures the number of egregious velocity errors and gives an indication about how robust the system is to producing velocity outliers.

C. Performance Comparison

We compare our learned method for joint association and tracking to the classical association methods described in Section II-C, due to their widespread use in filter-based tracking for real-time systems. We evaluate all methods on unimodal (LiDAR Only) and multimodal (LiDAR + Camera) configurations. All methods use the same IMM tracker. The results are detailed in Table I. Our proposed SDVTracker significantly improves system performance over other methods for both sensor modalities. For the LiDAR only system, we see improvements such as a 16% reduction in MOTVE, a 6.23% reduction in ID switches and a 2% reduction in false positives, over the next best method. Mahalanobis association has the best MOTP by 0.09 cm, but does not translate to

TABLE I

COMPARISON OF TRACKING METHODS ACROSS MULTIPLE SENSOR MODALITIES

Sensing Modalities	Method	MOTA ↑	MOTVO ↓		MOTVE ↓		FP ↓	FN ↓	IDSW ↓	MOTP ↓	MT ↑	ML ↓	Frag ↓
			Ped	Bike	Ped	Bike							
LiDAR	IoU-based Association	67.6486	3.572	2.377	0.170	0.287	394840	510918	44855	0.3527	0.389	0.164	39385
	L2 Association	67.9379	3.204	2.373	0.158	0.281	391298	**508561**	42092	0.3519	0.390	0.163	38850
	Mahalanobis Association	68.4670	2.956	2.041	0.157	0.271	370060	516321	37788	**0.3466**	0.386	0.165	40026
	SDVTracker (Ours)	**68.9816**	**2.199**	**1.633**	**0.131**	**0.248**	**362560**	510970	**35433**	0.3475	**0.391**	**0.162**	**38438**
LiDAR + Camera	IoU-based Association	66.5809	4.236	2.549	0.192	0.295	416723	503642	51731	0.3586	0.384	0.167	43417
	L2 Association	68.1027	3.303	2.334	0.162	0.294	386467	**497147**	43991	0.3554	**0.388**	0.163	40572
	Mahalanobis Association	68.6031	3.056	2.118	0.160	0.287	366913	504202	39521	0.3498	0.385	0.165	41251
	SDVTracker (Ours)	**69.4405**	**2.204**	**1.827**	**0.133**	**0.268**	**346651**	504744	**33118**	**0.3485**	**0.388**	**0.162**	**40008**

TABLE II

EFFECT OF LEARNING JOINT TRACKING AND ASSOCIATION

Network	IMM	Learning State	MOTA ↑	MOTVO ↓	MOTVE ↓	IDSW ↓
MLP	✓		69.2221	2.448	0.1446	37594
MLP	✓	✓	69.3863	2.385	0.1413	34698
LSTM	✓		69.2877	2.428	0.1419	35862
LSTM		✓	69.3971	**2.240**	0.1528	34031
LSTM	✓	✓	**69.4405**	2.292	**0.1393**	**33118**

TABLE III

EFFECT OF LEARNING PROBABILITY AND SCORE

Association Output	MOTA ↑	MOTVO ↓	MOTVE ↓	IDSW ↓
Probability Only	69.1837	2.544	0.1466	35419
Score Only	69.3618	2.551	0.1448	39461
Probability and Score	**69.4405**	**2.292**	**0.1393**	**33118**

better velocity estimates. This further demonstrates the need of metrics that measure higher order states when evaluating object tracking in 3D.

Furthermore, as more sensors are added to the system, we see an improvement in the overall MOTA and false negatives of methods. However, this comes at the cost tracking more objects, increasing the absolute number of velocity outliers. We show that our learned methods can better incorporate new sensor observations by reducing velocity outliers by 17%, ID switches by 16% and false positives by 5%. While Mahalanobis association sees a degradation in performance by around 3.3%, our learned method sees an increase in velocity outliers by 0.2%, all the while tracking more objects.

D. Impact of Recurrent Networks

We implement two learned network architectures for our learned association and tracker. For the recurrent network, we train on truncated sequences of length 20. We compare the performance of a Recurrent Neural Network (RNN) to a feedforward Multi-Layer Perceptron (MLP) in Table II. While both networks outperform classical association methods, we see a small increase in performance with the recurrent network.

E. Ablation on Joint Association-State Estimation

To understand the impact of jointly learning association and state estimation, we trained a recurrent and a feedforward network with and without including state estimation learning as a model output. The results are outlined in Table II. We see that regressing the state information improves performance for both network architectures. Further, we investigate how the model's learned state compares with the filtered IMM state. We see that while the model's learned state produces fewer velocity outliers, its average velocity and MOTA are worse compared to using the IMM, which motivates our hybrid method.

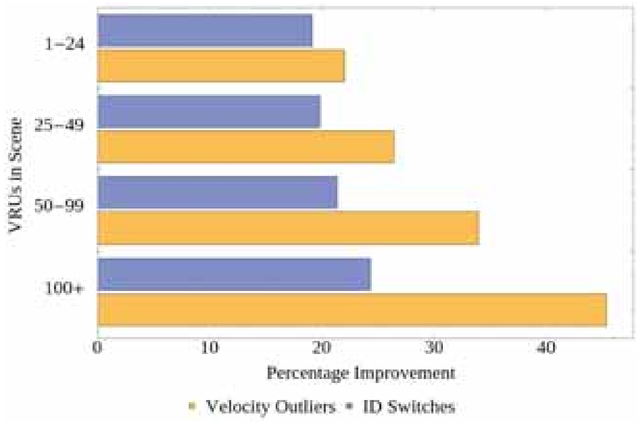


Fig. 5. As the number of pedestrians in a scene grows, our method is increasingly more effective at reducing velocity outliers than engineered methods. Analysis was performed on over 900 scenes bucketed by the number of pedestrians across a 25s interval, with each bucket including at least 20 scenes.

F. Ablation on Score Regression

We evaluate the effectiveness of learning both an association probability and a score, as discussed in Section III, in Table III. For the probability only model, we break ties between candidate detections based on the higher probability. For the score only model, we considered all scores below 0.1 as candidate associations. Based on the results, we see that neither breaking ties with the probability or thresholding based on the score perform better than explicitly learning a probability and a score.

G. Impact of Pedestrian Density

In dense crowds, a mis-association can cause a tracked object to have poor velocity estimates, which degrades system performance. Fig. 5 examines the performance of SDVTracker as the number of pedestrians in a scene is increased in terms of ID switches and velocity outliers. As pedestrian density increases, our proposed method performs

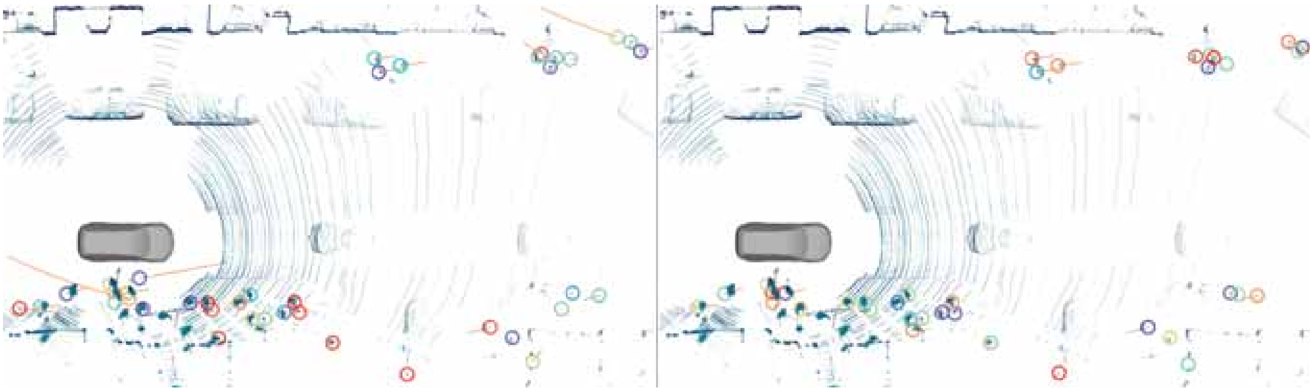


Fig. 6. (left) Classical Mahalanobis association and tracking. (right) SDVTracker, our system for learned association and tracking, which shows fewer velocity outliers. Circles represent tracked VRUs and orange vectors represent velocity estimates. See attached supplemental material for video versions.

better than hand-engineered association on both metrics. In scenes with 100+ pedestrians, the learned model reduces poor velocity estimates by 45%, demonstrating our learned model approach scales better than classical methods.

H. *Runtime Performance*

We show the runtime performance of the system in Fig. 7, evaluated on a four core Intel i7 CPU and a NVIDIA RTX 2080Ti GPU. We see that model runs under 5 ms for 500 actors on a CPU and under 3 ms on a GPU. It is interesting to note that for scenes with less than 100 VRUs, it is faster to run on CPU than using a dedicated GPU.

I. *Qualitative Performance*

Fig. 6 shows representative output of the classical Mahalanobis association and tracking compared to SDVTracker on a typical scene with VRUs. We see fewer velocity outliers, which yields better self-driving vehicle performance. Please refer to the provided supplemental video to see the SDVTracker in operation.

V. Conclusion and Future Work

We presented SDVTracker, a method for learning multi-class object-detection association and motion state estimation. We demonstrate that this algorithm improves tracking performance in a variety of metrics. In addition, we introduce new tracking metrics important in self-driving applications that measure the quality of the velocity estimates and show that SDVTracker significantly outperforms the compared methods. Furthermore, we demonstrate that SDVTracker generalizes to multiple sensor modalities, increasing recall with the addition of the camera sensing modality. Finally, we show this method is able to handle scenes of 100 actors under 2.5 ms, making it suitable for operation in real-time applications.

The performance of the learned state obtained directly from the LSTM was similar to the one obtained by the IMM, opening a door for new experiments to potentially remove the IMM from the algorithm while maintaining the performance. We plan to also augment the algorithm to learn the object life policy, controlling when to birth new objects and reap old ones. Finally, we further plan to extend SDVTracker by adding additional sensors, such as RADAR, to the system.

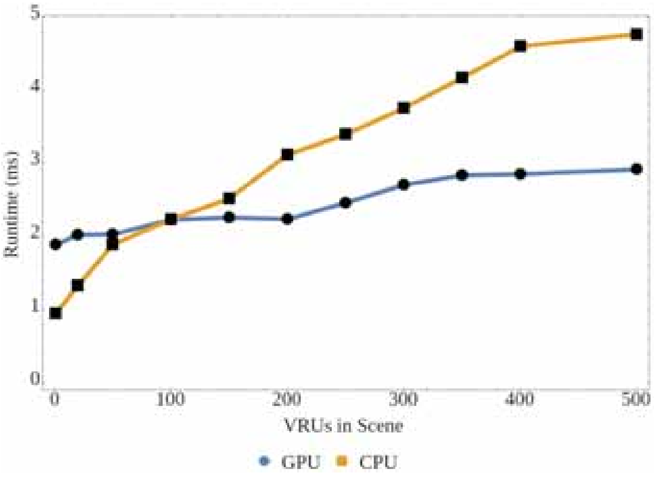


Fig. 7. Model inference runtime on CPU and GPU as a function of the number of actors in a scene. The model scales approximately linearly with the number of actors and for a typical scene with 100 actors runs under 2.5 ms on CPU.

VI. Acknowledgements

We would like to acknowledge and thank several people at Uber ATG who have made this research possible. We thank Blake Barber, Carl Wellington, Chengjie Zhang, David Wheeler, Gehua Yang, Kyle Ingersoll, Narek Melik-Barkhudarov, and Ralph Leyva for their support.

References

[1] W. H. O. D. of Violence, I. Prevention, W. H. O. Violence, I. Prevention, and W. H. Organization, *Global status report on road safety: time for action*. World Health Organization, 2009.

[2] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[3] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8454–8462.

[4] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[5] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[6] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.

[7] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 300–311.

[8] H. Farazi and S. Behnke, "Online visual robot tracking and identification using deep LSTM networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6118–6125.

[9] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," 2019.

[10] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[11] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, 2016.

[12] P. S. Maybeck, *Stochastic models, estimation, and control*. Academic press, 1982.

[13] B. Allotta, A. Caiti, R. Costanzi, F. Fanelli, D. Fenucci, E. Meli, and A. Ridolfi, "A new auv navigation system exploiting unscented kalman filter," *Ocean Engineering*, vol. 113, pp. 121–132, 2016.

[14] A. Genovese, "The interacting multiple model algorithm for accurate state estimation of maneuvering targets," 2001.

[15] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybrid-boosted multi-target tracker for crowded scene," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2953–2960.

[16] C.-H. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 685–692.

[17] S. Kim, S. Kwak, J. Feyereisl, and B. Han, "Online multi-target tracking by large margin structured learning," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 98–111.

[18] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4705–4713.

[19] P. Lenz, A. Geiger, and R. Urtasun, "FollowMe: Efficient online min-cost flow tracking with bounded memory and computation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4364–4372.

[20] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[21] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6951–6960.

[22] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 588–599, 2019.

[23] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[24] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.

[25] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[26] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[27] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[29] S. Song, Z. Xiang, and J. Liu, "Object tracking with 3d lidar via multi-task sparse learning," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2015, pp. 2603–2608.

[30] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[31] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[32] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

Situation Awareness at Autonomous Vehicle Handover: Preliminary Results of a Quantitative Analysis

Tamás D. Nagy[1,2], Dániel A. Drexler[1], Nikita Ukhrenkov[1], Árpád Takács[1] and Tamás Haidegger[1,3]

***Abstract*— Enforcing system level safety is a key research domain within self-driving technology. Current general development efforts aim for Level 3+ autonomy, where the vehicle controls both lateral and longitudinal motion of the dynamic driving task, while the driver is permitted to divert their attention, as long as she/he is able to react properly to a handover request initiated by the vehicle. Consequently, situation awareness of the human driver has become one of the most important metrics of handover safety. In this paper, the preliminary results of a user study are presented to quantitatively evaluate emergency handover performance, using custom-designed experimental setup, built upon the Master Console of the da Vinci Surgical System and the CARLA driving simulator. The measured control signals and the questionnaire filled out by participants were analyzed to gain further knowledge on the situation awareness of drivers during handover at Level 3 autonomy. The supporting, custom open-source platform developed is available at `https://github.com/ABC-iRobotics/dvrk_carla`.**

***Index terms*—Autonomous Vehicle Safety, Self-driving, Situation Awareness, Driving Simulator, Hand-over.**

I. Introduction

Autonomous driving technologies are on the rise worldwide, aiming to increase road safety in general. However, significant system and human failures have happened in the near past, indicating that the underlying technology and regulations are still just evolving [1]. The Society of Automotive Engineers (SAE) provided the most recognized scale for the levels of automation in the case of self-driving [2], a classification that is often used in different research domains as well [3], [4]. These Levels of Autonomy are:

- L0: no autonomy
- L1: user assistance
- L2: partial automation
- L3: conditional automation
- L4: high automation
- L5: full automation.

At L3 (conditional automation), most of the essential driving functions are automated, however, the driver should be ready to take control whenever it is necessary. Hazardous situations are typical sources of this transfer of control, when the automated system cannot handle the situation, and thus it notifies the user to resolve it. In the case of L3, safety considerations are crucial: due to the fact that most of the functions are automated, the driver can easily be distracted, unfocused and bored, while a smooth transfer of control requires constant attention from the user. Furthermore, drivers usually over-trust the system, causing lower level of Situation Awareness (SA) [1], [5]. One solution for this problem chosen by manufacturers is to implement higher level of automation directly (L4+), without these restricting conditions. Another, technically more feasible approach is to maintan high SA; the driver has a constant task to perform, such as handling the pedals solely, while it means retrogression in technology.

In driving automation, the term “handover” refers to taking back the control from the vehicle, and “takeover” (time) indicates the necessary timeframe in witch it actually happens [6]. Takeover is typically between 1.9 and 25.7 seconds in non-critical cases, however, it may get prolonged under critical conditions [7]. Takeover can be estimated from a control system model introduced in [8].

Situation Awareness is a key factor of driving safety (especially at L2 and L3). SA is defined on 3 levels based on the cognitive understanding of the (past–present–future) environment [9], [10]:

- **Level 1 SA**: Perception of the environment;
- **Level 2 SA**: Comprehension of the current situation;
- **Level 3 SA**: Projection of future status.

SA can be categorized into the following classes: spatial (locations), identity (salient objects), temporal, goal and system awareness (Fig. 1).

In this paper, we introduce an SA experiment, which examines the handover in emergency situations. In order to simulate these emergencies, we used a widely available driving simulator, CARLA[1] and the Master Console of the da Vinci Surgical System (Intuitive Surgical Inc., Sunnyvale, CA). We studied seven subjects’ handover performance under critical conditions.

II. Experimental Setup

The da Vinci Surgical System was originally developed for the purpose of robot-assisted minimally inva-

The research presented in this paper was carried out as part of the EFOP-3.6.2-16-2017-00016 project in the framework of the New Széchenyi Plan. This work was partially supported by ACMIT (Austrian Center for Medical Innovation and Technology), which is funded within the scope of the COMET (Competence Centers for Excellent Technologies) program of the Austrian Government. T. D. Nagy and T. Haidegger are supported through the New National Excellence Program of the Ministry of Human Capacities. T. Haidegger is a Bolyai Fellow of the Hungarian Academy of Sciences.

[1] Antal Bejczy Center for Intelligent Robotics, Óbuda University, Budapest, Hungary, {`tamas.daniel.nagy, arpad.takacs, tamas.haidegger`}`@irob.uni-obuda.hu`, `daniel.drexler@nik.uni-obuda.hu`, `nikita.ukhrenkov@gmail.com`

[2] Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University, Budapest, Hungary

[3] Austrian Center for Medical Innovation and Technology (ACMIT), Wiener Neustadt, Austria

[1] http://carla.org/

sive surgery [11]. Its human–machine interface is versatile enough to be used for the purpose of self-driving handover experiments. The head-in type stereo display is an excellent tool to control and monitor the driver's attention—just like the surgeon's attention in the conventional, clinical use. When the driver's head is not inserted, they are not able to see the simulation, and likewise, when their head is inserted, no external visual disturbances may pass into their field of view. Furthermore, thanks to the built-in photogates, the insertion of the head into the display area can be easily detected. The Master Tool Manipulators (MTMs) of the da Vinci Master Console, as well as the foot pedals were tailored to offer similar functionality to the steering wheel and foot pedals of a car [12].

The implemented system (Fig. 2) was built upon two mayor open-source software components: the Da Vinci Research Kit (DVRK) [13] and the CARLA Simulator [14]. The MTMs of the Master Console mimic the behavior of a steering wheel, relying on the impedance control built into the DVRK; the built-in head sensor is also interfaced to the control PC through the DVRK platform; the foot pedals extended with Hall effect sensors are connected using an Arduino board (Arduino Co., Somerville, MA) [15], sending the measured values to a Robot Operating System[2] (ROS) environment; the stereo display is connected to the PC using DVI interface. The control PC runs the *cisst-component* [16] to interface DVRK—and so do the MTMs and the head sensor—to ROS and the CARLA server, responsible for the simulation. Moreover, a ROS node sets the gains of the impedance control dynamically, and a CARLA client forwards the control values to the CARLA server and sends the stereo video stream to the displays.

The MTMs of the da Vinci are programmable using the open-source DVRK platform [16], which is based on the highly modular ROS, used widely in robotics research [17]. At the tips of the MTMs, 3D printed wheel segments were fixed (Fig. 3). The motion of this DVRK steering wheel is restricted to a circular trajectory around a virtual center point using the built-in impedance control of the DVRK [12], and the steering angle value is interfaced to the CARLA client over ROS (Fig. 2).

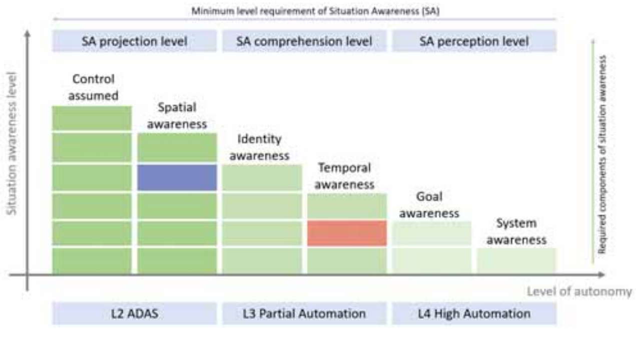


Fig. 1. Hierarchical representation of Situation Awareness (SA) in self-driving vehicles. For every level of autonomy(L2 Advanced Driver-Assistance System (ADAS), L3 Partial Automation and L4 High Automation), the quantitative metrics must fulfill the requirements for each category.

The usage of the foot pedals of the da Vinci Master Console for the driving experiments was an obvious choice. However, those pedals offer simple binary output by default. To get continuous reading, the pedals were completed with Hall effect sensors and small-sized magnets, connected to the PC using an Arduino board, serving as accelerator and break pedals. The sensor values were read using the `rosserial_arduino` package, and were forwarded through ROS topics towards the CARLA client (Fig. 2).

The two displays of the da Vinci—serving as stereo display pair—have been replaced with LCD screens to enhance image quality—, which is a commonly used enhancement of the DVRK platform. These screens were connected to the PC over DVI to provide the stereo video stream to the driver. The head-in type display allowed attention control for the drivers, as they were not aware of the environment and the simulator at the same time. Moreover, using the built-in photogates of the console, the insertion of the driver's head was also monitored. The signal of the photogates was forwarded to a ROS topic through one of the DVRK controllers (Fig. 2).

The CARLA Simulator was chosen to be used in the experiment; this open-source driving simulator is used widely in the research of autonomous driving, furthermore, it offers built-in scenarios, autopilot and ROS communication [14]. The CARLA Server offers the core of the simulation, while a CARLA client forwards the steering angle and pedals values form ROS using Remote Procedure Calls (RPC). Moreover, it defines the two cameras to ensure stereo vision (Fig. 2.), forwarding the video stream to the display of the da Vinci Master Console.

III. Experimental Protocol

In the experiments, it was our aim to model hand-over processes at L3 autonomy during emergencies. Each individual experiment was divided into 8 successive scenarios, none of the subjects participated in more than one experiment. Before each experiment, the subjects had one minute to practice driving in the simulator.

Every scenario started by the car driving autonomously, while the subject was instructed to type a text message on a smartphone, and not to insert her/his head into the simulator display nor pay attention to it. After 40–60 seconds of autonomous driving the system raised an emergency audio alarm and yielded the control to the human subject. This time delay was randomly chosen for the 8 scenarios at the beginning of the experiment, and was the same for each subject. This way, despite subjects would not expect the alarm at the same time instant, the results remained comparable between subjects. Then, the subjects had to take control of the vehicle and tried to solve the traffic situation. The subjects were also instructed that unnecessary braking (e.g., in the case of false alarm, see below) was unwanted and inflicted penalty. Each of the the 8 emergency scenarios happened at the same location on the simulation's map, with

[2] https://www.ros.org/

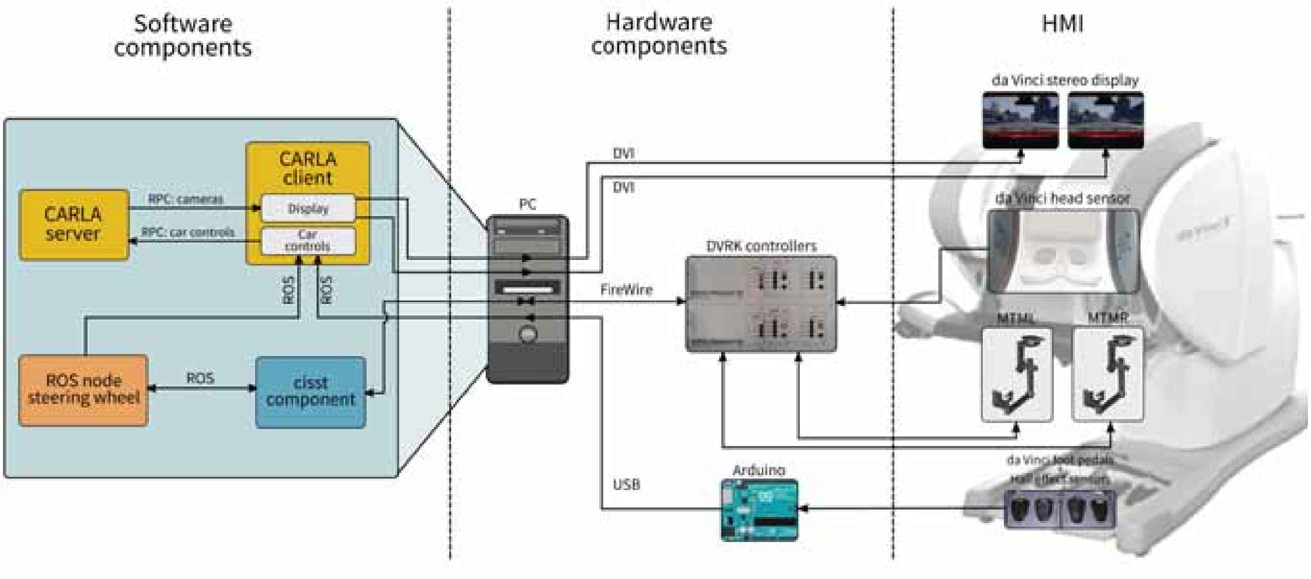


Fig. 2. Block diagram of the experimental setup. The display, the head sensor, the input manipulators and the pedals of the da Vinci Master Console are used to create a handover simulation user interface. The Da Vinci Research Kit is used for control, while the setup is interfaced to the CARLA Simulator via ROS components.

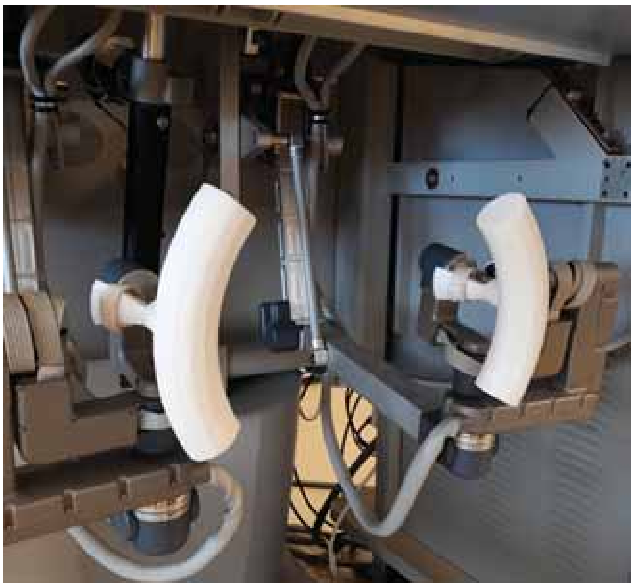


Fig. 3. The da Vinci MTMs with the 3D printed steering wheel segments, push-fitted and fixed by the built-in hook-and-loop fasteners. Using impedance control, the arms are mimicking the behavior of steering wheels, and rotate around a virtual axis.

the combination of the two states of the following three conditions:

1) **True/False alarm:** A pedestrian was involved in the emergency in all of the designed scenarios. In the case of the true alarm, the pedestrian stepped in front of the vehicle from behind a vending machine (Fig. 4), and the car was about to hit him. In the case of the false alarm, the pedestrian was moving on the sidewalk, parallel to the road. This case could have also been done without a pedestrian, however, we decided to leave the pedestrian in the scenario because his motion could also trigger braking at some of the subjects. The audio alarm was always raised three seconds before reaching the pedestrian’s location;
2) **Car coming from front/No car coming from front:** To make the scenarios more challenging, opposing traffic was added to the scenario at the location of the emergency at some of the scenarios. In the case of no car coming from the opposite lane, there were no other vehicles on the road;
3) **Clear weather/Heavy rain:** To change visual conditions, the weather was also changed between scenarios.

Using the three varying conditions above, the following order of scenarios was compiled (the same for each subject):

1) True alarm, No car, Clear weather;
2) False alarm, Car coming from opposite lane, Clear weather;
3) True alarm, Car coming from opposite lane, Heavy rain;
4) True alarm, Car coming from opposite lane, Clear weather;
5) False alarm, Car coming from opposite lane, Heavy rain;
6) False alarm, No car, Heavy rain;
7) True alarm, No car, Heavy rain;
8) False alarm, No car, Clear weather.

In parallel to the scenarios on the simulator, the subjects were also asked to fill in a questionnaire. Before the introductory driving practice and the scenarios, they were asked to read and agree to a consent form; the data gathered was completely anonymous. Afterwards, some general questions were asked regarding their age and driving experience. Following

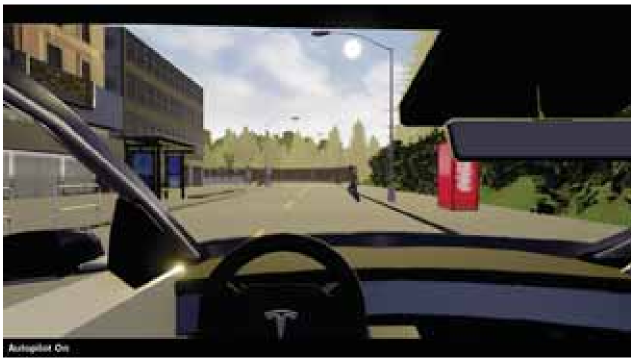


Fig. 4. Screenshot of the simulation in one of the emergency scenarios. The pedestrian is stepping down to the road ahead the vehicle from behind a vending machine; the weather is clear with good visual conditions and there is no traffic on the road.

each scenario, questions regarding the simulated event and the details of the environment were asked to gain further information on their SA. Furthermore, after each scenario, they were asked to evaluate their own reaction on a scale 1–5. See the details of the questionnaire in Section IV.

IV. Results

We measured the SA of the participants by asking questions about their surroundings. They got 1 point for the good answer, 0 point for neutral answer (I do not know) and -1 point for a wrong answer. There was a specific case when they were asked about the direction of travel after the accident scene, where straight and left was also a good answer, although the road turned to left in a short distance; in this case straight was also accepted as a good answer with 0.5 point. The evolution of the SA along the scenarios are shown in Fig. 5 for all the participants.

We measured the takeover time as the difference between the time of the *handover request* (alarm sound) and the time of the first physical reaction (large change in steering wheel angle or break pedal operation) after the handover request. The car switched to manual drive as the handover was initiated, thus by the time the participants looked into the display, the car already started drift off the lane. As a result, an immediate intervention was always necessary in all the scenarios. The values of takeover times for each participant and each scenario are shown in Fig. 6.

The takeover time for each scenario is shown in Fig. 7, using a compact box plot. The circles are outlier data, dotted circles indicate the median. The thick lines show the range, where the second and third quadrant of the data are, and the thin lines show the range of other non-outlier data. One can observe a slight decrease in the takeover time medians as the scenario index increases, which may imply that as the subjects gained SA, thus their handover performance increased.

The increase of SA can be observed in the slight increase of general satisfaction in Fig. 8. The figure shows how the mean satisfaction increased (based on the survey) during different scenarios. The satisfaction for each scenario was acquired from the questionnaire, where the participants were asked to rate their own reaction on a scale from 1–5 (1–bad, 5–excellent). The SA was also checked by asking questions about the surroundings, which become more accurate as the participants moved forward in the experiments. Fig. 8 shows that polling the self-satisfaction might be indicative of the SA of the subject.

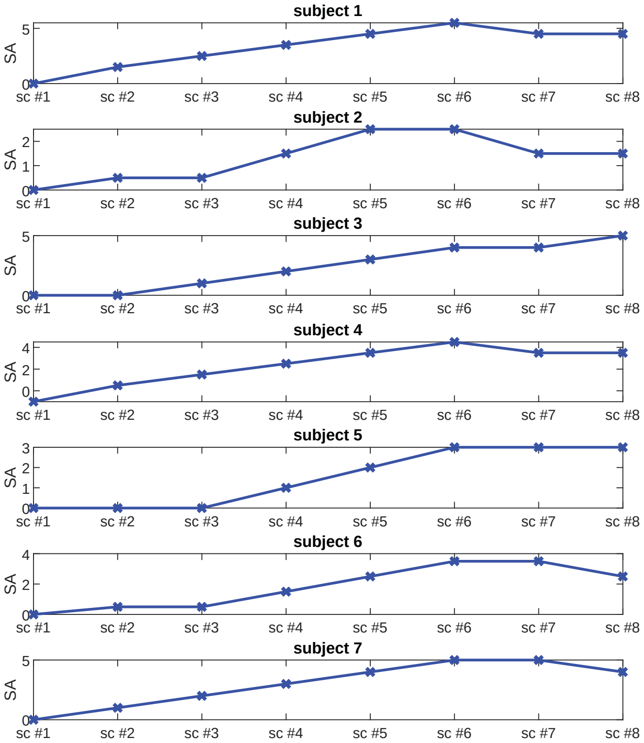


Fig. 5. The evolution of Situation Awareness (SA) of the participants along the scenarios.

The mean satisfaction of the subjects is shown versus their mean takeover time in Fig. 9. The subjects could be divided into three groups intuitively. The first group consisted of subject 7, who had the smallest mean takeover time, and the largest satisfaction. The second group was composed of subjects 1 and 3, who had the larges takeover time, but still high satisfaction. The third group was composed of subjects 2,4,5,6, who had relatively small takeover time, but also small satisfaction. This shows that general satisfaction does not correlate with the mean takeover time.

Although Fig. 9 shows that the mean satisfaction does not correlate with mean takeover time, Fig. 8 and the answers from the questionnaire show that mean satisfaction correlates with SA. This may imply that SA has does not correlate with mean takeover time, but this implication is wrong. Subject 1 had large mean takeover time, however, this is because of the large takeover time in scenario #1, and as the SA of subject 1 increases, the takeover times decreases (Fig. 6). For subject 3, the takeover time was large for the first and the last scenarios, but there is a weak decreasing tendency in the takeover times, which may be connected to increasing SA. The large takeover times can be associated with the unique

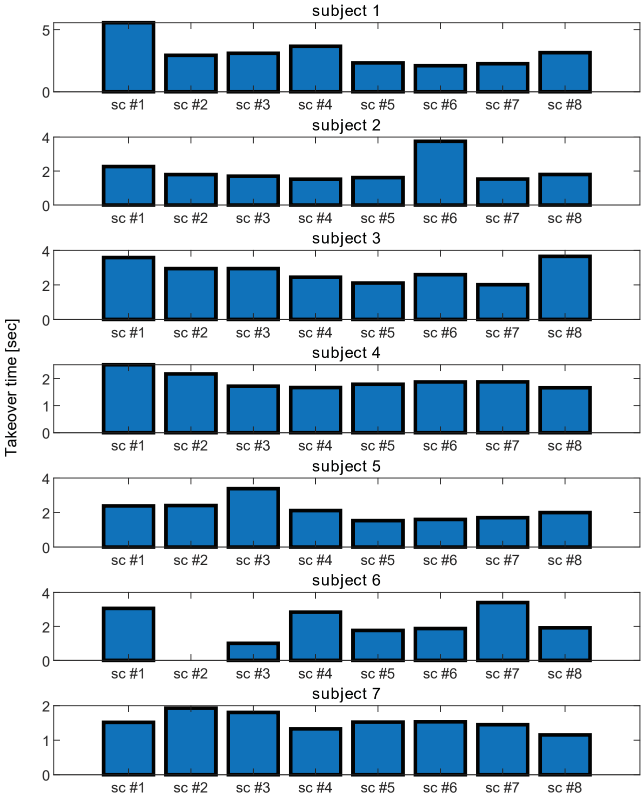


Fig. 6. The takeover times of the participants in the 8 scenarios.

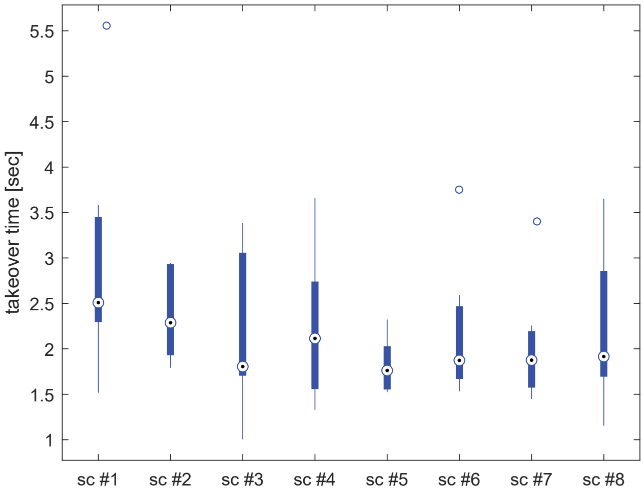


Fig. 7. The takeover times in the 8 scenarios depicted in a compact box plot: circles show outliers, dotted circles are the medians, the thick lines show the ranges where the second and third quadrant of the takeover times are (25–75%), and thin lines show the range of all the other takeover times in the current scenario.

personal capabilities of subject 3. This alludes that using plots like Fig. 9 for evaluation of a handover system may be misleading due to the different abilities of the subjects.

V. Conclusion and Future Work

In this paper, a preliminary user study was presented based on our objective human performance assessment platform, built on DVRK and CARLA Simulator. The system was used to evaluate the handover process during emergency situations of autonomous driving at L3. The user trial, including a questionnaire, was conducted on 7 test subjects, in 8 successive scenarios. We found the resulting takeover times on the simulator to be concordant with the values described in the literature, which projects that our results in the simulated environment can be translated into real life situations. It was observed the slight decrease of takeover time over the successive scenarios, which may imply the increasing Situation Awareness of the test subjects. The SA scoring, based on the questionnaire, shows an increasing tendency during the scenarios, that, similarly to the takeover time, implies the gaining of SA of the subjects. However,

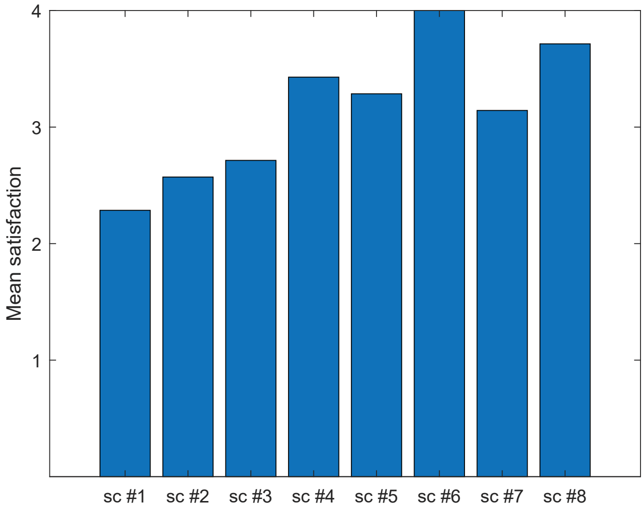


Fig. 8. The mean satisfaction (averaged for all the participants) for each scenario. Satisfaction was asked from the participants after each scenario, they rated their performance on a scale of 1–5 (1–bad, 5–excellent).

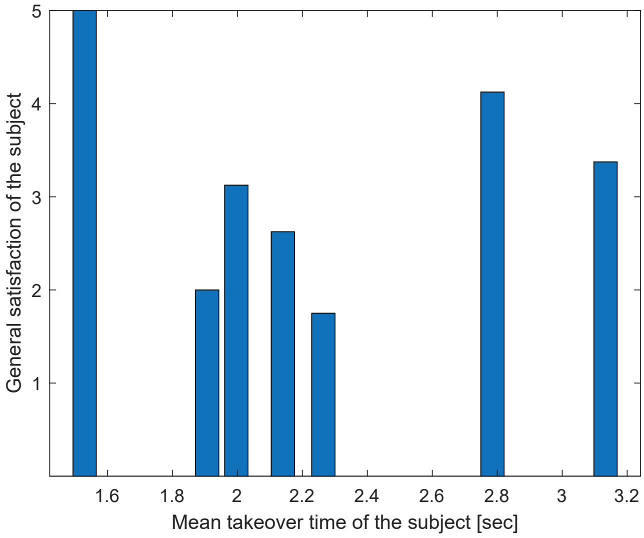


Fig. 9. The mean satisfaction of the subjects and their mean takeover times. Repeated scenarios' outcome was averaged for the same subject. Two subjects presented a certain self-biased behavior during the experiment. One subject was arguably best.

the results of the rating of the subjects' own performance from the questionnaire, which should also be closely related to SA, do not seem to correlate with the takeover time. This contradiction is possibly originating from the subjective nature of this question of the questionnaire. In the upcoming user studies, with a greater number of subjects and improved scenarios, these questions might be answered with higher certainty.

The open-source implementation of the platform is available on GitHub at `https://github.com/ABC-iRobotics/dvrk_carla`.

References

[1] V. A. Banks, K. L. Plant, and N. A. Stanton, "Driver error or designer error: Using the Perceptual Cycle Model to explore the circumstances surrounding the fatal Tesla crash on 7th May 2016," *Safety Science*, vol. 108, pp. 278–285, Oct. 2018.

[2] "Taxonomy and Definitions for terms related to driving automation systems for on-road motor vehicles (J3016)," Society for Automotive Engineering (SAE), Tech. Rep., 2016.

[3] T. Haidegger, "Autonomy for Surgical Robots: Concepts and Paradigms," *IEEE Trans. on Medical Robotics and Bionics*, vol. 1, no. 2, pp. 65–76, 2019.

[4] D. A. Drexler, A. Takacs, D. T. Nagy, and T. Haidegger, "Handover Process of Autonomous Vehicles – technology and application challenges," *Acta Polytechnica Hungarica*, vol. 15, no. 5, pp. 101–120, 2019.

[5] V. A. Banks, A. Eriksson, J. O'Donoghue, and N. A. Stanton, "Is partially automated driving a bad idea? Observations from an on-road study," *Applied Ergonomics*, vol. 68, pp. 138–145, Apr. 2018.

[6] P. Morgan, C. Alford, and G. Parkhurst, "Handover issues in autonomous driving: A literature review," University of the West of England, Bristol, Project Report, 2016.

[7] A. Eriksson and N. A. Stanton, "Takeover Time in Highly Automated Vehicles: Noncritical Transitions to and From Manual Control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 59, no. 4, pp. 689–705, June 2017.

[8] C. Gold, R. Happee, and K. Bengler, "Modeling take-over performance in level 3 conditionally automated vehicles," *Accident Analysis & Prevention*, vol. 116, pp. 3–13, July 2018.

[9] Endsley, M.R., "Situation awareness global assessment technique (SAGAT)," in *Proc. of the IEEE 1988 National Aerospace and Electronics Conference*, Dayton, OH, USA, 1988, pp. 789–795.

[10] Endsley, M.R., "Situation Awareness in Aviation Systems," in *Handbook of Aviation Human Factors, Second Edition*. CRC Press, Dec. 2009.

[11] G. Chrysilla, N. Eusman, A. Deguet, and P. Kazanzides, "A Compliance Model to Improve the Accuracy of the da Vinci Research Kit (dVRK)," *Acta Polytechnica Hungarica*, vol. 16, no. 8, Sept. 2019.

[12] T. D. Nagy, N. Ukhrenkov, D. A. Drexler, Á. Takács, and T. Haidegger, "Enabling quantitative analysis of situation awareness: System architecture for autonomous vehicle handover studies," in *Proc. of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Bari, Italy, 2019, pp. 914–918.

[13] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci® Surgical System," in *Proc. of the IEEE International Conference on Robotics and Automation*, Hong Kong, 2014, pp. 6434–6439.

[14] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[15] A. D'Ausilio, "Arduino: A low-cost multipurpose lab equipment," *Behavior Research Methods*, vol. 44, no. 2, pp. 305–313, June 2012.

[16] Z. Chen, A. Deguet, R. H. Taylor, and P. Kazanzides, "Software Architecture of the Da Vinci Research Kit," in *Proc. of the IEEE International Conference on Robotic Computing (IRC)*, Taichung City, Taiwan, 2017, pp. 180–187.

[17] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: An open-source Robot Operating System," in *Proc. of the ICRA Workshop on Open Source Software*, vol. 3, Kobe, Japan, 2009.

Towards Context-Aware Navigation for Long-Term Autonomy in Agricultural Environments

Mark Höllmann*, Benjamin Kisliuk*, Jan Christoph Krause*, Christoph Tieben*, Alexander Mock†, Sebastian Pütz†, Felix Igelbrink†, Thomas Wiemann*†, Santiago Focke Martinez*, Stefan Stiene*, Joachim Hertzberg*†

*DFKI Niedersachsen Lab
Plan Based Robot Control Group
Osnabrück, Germany
firstname.lastname@dfki.de

†University Osnabrück
Knowledge-based Systems Group
Osnabrück, Germany
firstname.lastname@uni-osnabrueck.de

***Abstract*—Autonomous surveying systems for agricultural applications are becoming increasingly important. Currently, most systems are remote-controlled or relying on a single global map representation. Over the last years, several use-case-specific representations for path and action planning in different contexts have been proposed. However, solely relying on fixed representations and action schemes limits the flexibility of autonomous systems. Especially in agriculture, the surroundings in which autonomous systems are deployed, may change rapidly during vegetation periods, and the complexity of the environment may vary depending on farm size and season. In this paper, we propose a context-aware system implemented in ROS that allows to change the representation, planning strategy and execution logics based on a spatially grounded semantic context. Our vision is to build up an autonomous system called Autonomous Robotic Experimental Platform (AROX) that is able to generate crop maps over a whole vegetation period without any user interference. To this end, we built up the hardware infrastructure for storing and charging the robot as well as the needed software to realize context-awareness using available ROS packages.**

***Index Terms*—Autonomous systems, Context awareness, Navigation, Path Planning**

I. Introduction

To successfully deploy autonomous vehicles for long-term autonomy in dynamic environments like agriculture, it is necessary to take the current application context into account. In agriculture, unlike classic indoor scenarious, the shape of the environment may change rapidly. Hence, solely relying on established gridmap-based solutions is not possible. Depending on the time of year, the state of the crops, and the field's layout, more specialized solutions are needed. On the other hand, established methods for simpler use cases are well tested and understood, and might still be able to solve specific sub-problems. To fuse the usage of classic algorithms and novel, application-specific methods, we propose to take the geometric and semantic context into account. More specifically, we present an autonomous system called AROX for crop monitoring, that is deployed on a real farm. In addition to the mobile robot, the system consists of a storage and charging facility placed next to the monitored fields. The storage and charging station provides the shelter for the AROX robot (cf. Fig 1). In this scenario, the robot's objective is to periodically take scans of a maize field without user interaction. To achieve this, we built a semantically annotated geometric environment model that is divided into different navigation zones. Each zone exhibits different surface properties that require specific navigation algorithms or parametrization. For example, while driving and maneuvering within the container with the charg-

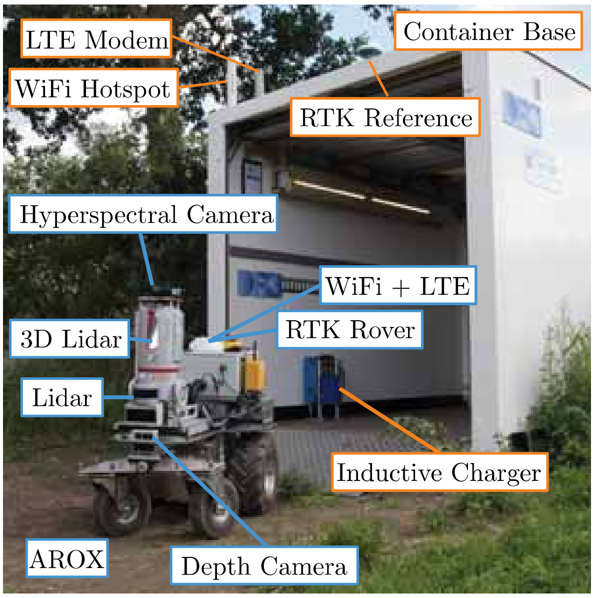


Fig. 1. Overview of the used hardware components on the AROX robot (blue) and storage container (orange) for autonomous long term crop monitoring.

The DFKI Niedersachsen Lab (DFKI NI) is sponsored by the Ministry of Science and Culture of Lower Saxony and the VolkswagenStiftung. Work by Wiemann and Focke Martinez is supported by the Federal Ministry of Education and Research within the framework of the BonaRes programme (grant number: 031B0684D). Work by Höllmann and Tieben is supported by the Federal Ministry of Food and Agriculture within the experimental field Agro-Nordwest project (grant number: 28DE103E18)

ing station, the robot will use classic gridmap based navigation and action planning algorithms for docking. While driving on the field, it shall use polygonal 3D environment maps for path planning and execution, which include trafficability and roughness estimations derived from the recorded sensor data as described in [1]. The main idea of this work is to use the most appropriate specialized planning and navigation method based on the semantic context given in the geometric environment model. Although such an zone-based approach is not completely novel conceptually, for future development we plan to focus more on the integration of semantics for reasoning. In this paper, we contribute the low level implementation of the technical aspects of such a semantic system for robust navigation based on a semantic environment map. We present the system components of the used surveying system and the implementation of context-awareness in ROS, and give a preliminary report on the current state of the already realized components of the planned system as well as a proof of concept for the proposed context-aware navigation scheme.

The remainder of this paper is organized as follows: First, we present state-of-the-art algorithms and environment representations used in our context. Sec. III presents the hardware infrastructure implemented on the actual farm. Sec. IV presents the implementation of the context-aware planning and navigation system. Sec. V presents the current state of implementation by means of exemplary application scenarious. The final section discusses the achieved results and shows the planned extensions of the proposed system.

II. Related Work

Previous works like the one by Marder-Eppstein et al. [2] describe the software components needed for robust robot navigation. Many of them can be found in various implementations collected in the *ROS Navigation Stack*. The usual setup for such an autonomous system usually consists of an environment map, which is often some kind of occupancy gridmap [3], means for localization within such a map like *AMCL* [4] for indoor scenarios or the *robot_localization* package by *Charles River Analytics* in outdoor contexts to fuse *GNSS* position data with IMU and the odometry measurements. Path planning and execution are typically divided into global and local planning. Global planning typically involves the static information about distant parts of the environment encoded in a topological graph. Local navigation copes with the dynamic environment within the robots sensory horizon. For each of those components, various approaches and implementations already exist in ROS. The system presented in this paper is similar in that respect to these conventional approaches, but does not rely on fixed implementations of the single planning and execution components.

A. Flexible Navigation

An example for such a more or less static execution stack is *move_base*. However, more recently, Pütz et al. [5] presented a flexible navigation framework called *Move Base Flex* (MBF). It has been introduced specifically to address the highly dynamic and heterogeneous nature of navigation contexts, which may require highly different approaches to solve the aforementioned sub-problems. The main benefit of using *move_base_flex* is that it provides the possibility to easily switch between different navigation approaches. It uses the same software interfaces as `move_base`, but allows for online reconfiguration and replacement of planning and recovery movement strategies via a user implemented *SMACH* state machine [6]. In contrast to earlier approaches published by Conner et al. [7], it avoids multi-instantiation of core components and thus does not suffer from such computational overhead.

B. Semantic Context Mapping

Generally, maps for localization and navigation strictly rely on spatial information. Beyond that, semantic meaning was introduced early to enhance pure geometrical or topological representations. Semantic maps add semantic attributes to the geometric representation that allow to reason about the environment. First approaches proposed a layered architecture to fuse the different information domains [8], [9], resulting in various modern forms of introducing context information to maps by anchoring knowledge about objects, areas, and spaces to the geometrical robotic map representation. This context information is used to improve the performance in various robotic tasks such as object detection, object retrieval, task planning, navigation and more [8], [10]–[12].

However, these approaches regarding object localization are confined to *small-scale* spaces and are usually limited to small semantic domain models. In a more general and formal approach, Lang and Paulus define semantic maps as a form of hybrid map as defined by Buschka [13], combining geometrical representations of an environment with knowledge about the entities contained within the represented environment, their classes and attributes, and the relations between them in a way that allows inferencing [14].

Following this concept, we propose a way to handle heterogeneous navigation contexts derived from geometric layout and semantic labels. In our setup, we use that information to navigate on the farm and to switch between multiple, heterogeneous contexts requiring different localization methods, planning algorithms, local planners and recovery strategies by using a semantically annotated map and MBF.

III. (Hardware) Infrastructure

For long-term autonomy in agricultural environments, a base station is needed. In addition to the mobile robot platform, this station stores required supply equipment and external computation hardware. Furthermore, the overall hardware must be protected from unauthorized access and extreme weather conditions. In our setup, a $3.7\,\mathrm{m} \times 2.55\,\mathrm{m} \times 2.1\,\mathrm{m}$ container is used as base station. The so-called RYTLE HUB is mobile, so it is possible to transport all parts as a ready-to-operate system. Access through two electrical sliding gates is authorized with Bluetooth beacons. Via a mechanically fold-out ramp, the robot can reach the inductive charging station. Power supply is provided by a battery system or common line voltage. This is required to operate the multi-band RTK GNSS reference station. For data exchange between base and robot, a high range WiFi network with 100 meter range is provided by the container. This is also used to transmit correction data from the RTK reference station to the rover. In addition to this WiFi network, both are equipped with a LTE modem to connect to the internet.

The AROX robot is a two-axes mobile platform based on an Innok Heros[1]. For pose estimation, sensor data from IMU, odometry and RTK GNSS are fused using the *robot_localization* ROS package. Besides these sensors for dead-reckoning, two laser scanners, located at the front and rear of the platform, are used for collision avoidance and localization. For crop monitoring and environment mapping, a high resolution 3D laser scanner with co-calibrated hyperspectral or RGB-camera is used. The entire equipment of the base station and the AROX robot is shown in Fig. 1.

IV. Context-aware navigation and planning

Our approach to deal with the rapidly changing and highly variable environment in agricultural environments is context-aware navigation and planning. Currently, we model the contextual data manually. The spatial dimension of each context is defined using geo-referenced polygons which can be associated with semantic labels. Additionally, we define specific waypoints, where the robot can switch from one context to another.

For example, one zone is the container. The corresponding semantic information is that the robot must navigate carefully and accurately due to the confined space. For this, the robot uses classic gridmap-based navigation and action planning. Another zone is the grass area in front of the container where the robot can navigate faster and less accurately. For this the robot uses a polygonal 3D environment map, that encodes the trafficability of the rough surfaces [1]. An example of such a waypoint is the gate that separates the container and the grass area as shown in Fig. 2.

A. Waypoint Server

The waypoint server takes the created zones and waypoints to build a topological graph. This graph includes all waypoints as vertices and inserts edges between all waypoints belonging to the same zone. Each waypoint is associated with both zones it connects. For each zone, the waypoint server stores the corresponding path planner, controller, recovery behaviors, and environment representation, and computes the locomotion costs for every edge in the graph. To query the waypoint server for the shortest path between pose A and pose B, both poses are first added as temporary vertices. Subsequently, edges are created to connect both of the new vertices to all other waypoints belonging to the same zones, and the locomotion costs for these edges are computed. If both poses belong to the same zone, a direct edge between them is also created. In this temporary extended graph, the shortest path from A to B is then calculated using a shortest path algorithm. An exemplary result is displayed in Fig. 2. The result of the query are the edges between pose A to B including the contextual dependent information of the traversed zones .

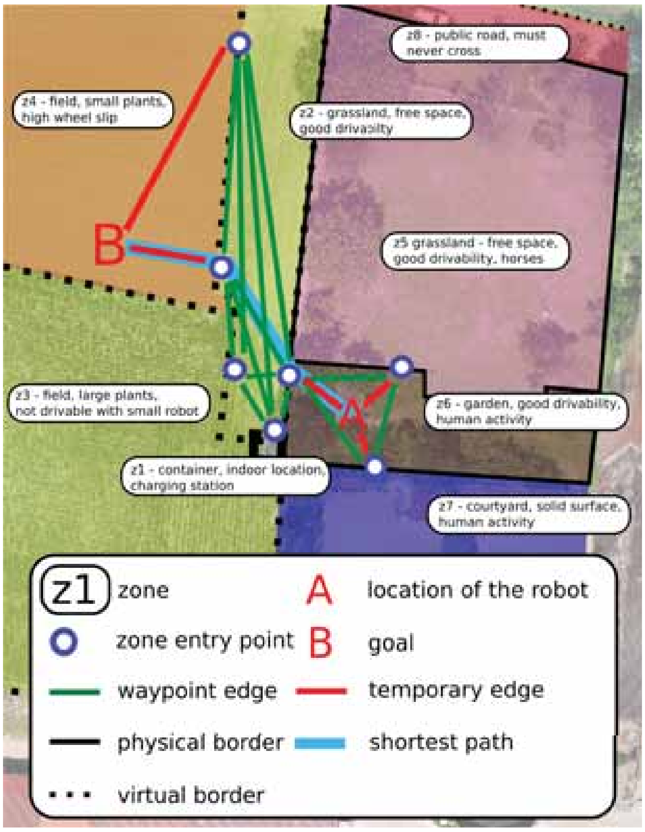

z8 - public road, must never cross
z4 - field, small plants, high wheel slip
z2 - grassland, free space, good drivabilty
B
z5 grassland - free space, good drivability, horses
z3 - field, large plants, not drivable with small robot
A
z6 - garden, good drivability, human activity
z1 - container, indoor location, charging station
z7 - courtyard, solid surface, human activity
z1 zone
A location of the robot
zone entry point
B goal
waypoint edge
temporary edge
physical border
shortest path
virtual border

Fig. 2. A schematic map with waypoints and semantic associations.

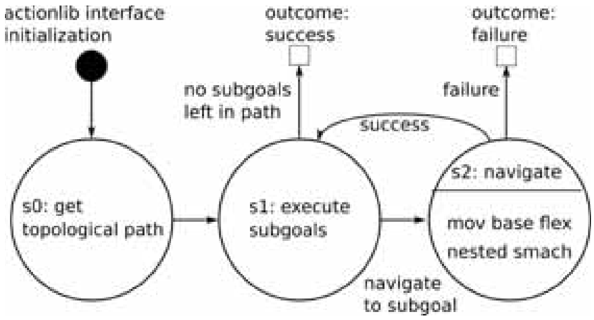


Fig. 3. The SMACH state machine for path decomposition.

B. Path Decomposition State Machine

To provide an abstract interface for this process, the Path Decomposition State Machine acts as an adapter and execution layer for the waypoint server and the navigation components. It is implemented as a *SMACH* hierarchical finite state machine within an *actionlib* wrapper to easily integrate into the ROS architecture of the robot as shown in Fig 3. Initially, it will be instantiated in *s0* by the action wrapper and given

[1] https://www.innok-robotics.de/en/products/heros

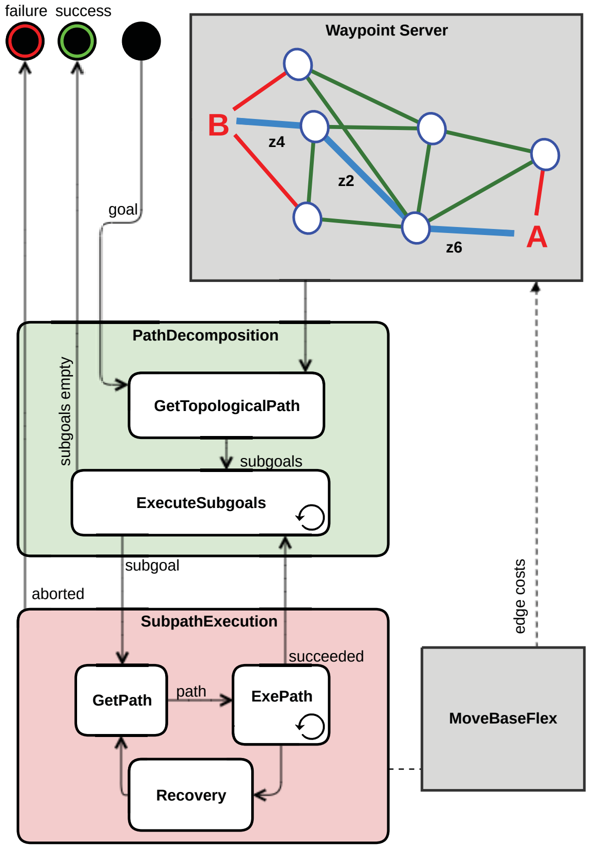


Fig. 4. Context-Aware Navigation state machine.

a goal pose which includes the geometric target pose and a robot coordinate frame known to the robot within its *tf-tree* coordinate frame. From there, it will request the current location of the robot, ask the waypoint server for the shortest path from this to the set goal, and transition to the execution state *s1*. The retrieved path consists of a list of sub-goals defined as tuples of geometric poses and parameters such as the local and global planner plugin to be used, and a list of recovery strategies. As long as the list of sub-goals is not empty, the first entry will be removed from the list and passed along the transition to the navigation state *s2*, which is a nested state machine interfacing directly to MBF. When this nested state machine terminates, the outcome will either be escalated to the action interface in case of failure, or it will transition back to the execution state *s1* where the next sub-goal will be passed or, in case the list of sub-goals is empty, terminate to a success outcome.

C. Sub-Path Execution

The actual movement of the robot to a sub-goal is executed by the sub-path execution state machine. For this purpose, the path-decomposition passes a sub-goal to the sub path execution, which is located in a zone together with the robot. This zone is associated with a path planner and controller names, as well as with a list of recovery behaviors as fallback strategies. In order to be able to flexibly execute and switch these sub-components of robot navigation, we use the MBF package *mbf_costmap_nav*, which provides a MBF navigation server based on the well known layered *costmap_2d* occupancy gridmap implementation. MBF is used as middle layer in between the higher level state machine and the low level navigation modules for planning, motion control, and recovering, which are implemented as plugins. It allows to load and run multiple plugins of the same type in a parallel fashion, e.g, with different configurations and different names. The *actionlib* interface of MBF allows to call path planners, controllers, and recovery behaviors with an associated name separately to support a high level of flexibility. According to the MBF interface, the sub-path execution consists of the three states *GetPath*, *ExePath* and *Recovery*. Using *GetPath*, MBF is called with the dynamically specified planner plugin name and the required start and goal pose parameters, as well as optional parameters such as the distance and angle tolerance to the goal pose. Based on this, MBF executes the corresponding path planning plugin and returns the computed path if the planning was successful, and a specified error code otherwise. The called path planner is using the underlying map representation to compute a path towards the given sub-target which has been defined as action goal for *GetPath*. This path is then used to call the controller within the *ExePath* state, in order to move the robot along this execution path. Using *ExePath*, MBF is called with a specified controller name and a list of poses defining the path to periodically execute the corresponding loaded controller / local planner plugin in order to move the robot towards the goal pose. The called navigation controllers manage the robot locomotion with different implemented strategies. This way, the controllers try to follow the given path while detecting dynamic obstacles. In case of error or failure, the MBF transitions to *aborted* and back-propagates the controller error code to the *SMACH* task level execution. Depending on the outcome, different strategies and recovery behaviors can be called in order to resolve the problem, e.g., by clearing the costmap, rotating to get an overview, waiting for an obstacle to pass, or moving backwards to escape from a possible collision or local dead end. However, when the robot reaches the sub-goal, the sub-path execution transitions to the terminal state *succeeded*. In cases of error, i.e., the controller failed, recovery behaviors which are domain-specifically associated with a certain zone, are executed by calling the MBF *Recovery* action with the corresponding recovery name. If the robot is able to free itself from a difficult situation with one of the recovery behaviors, the task level architecture resumes to the sub-path execution.

V. Applications

A. Demo Scenario

To show the capabilities of our planning system, we are currently building an outdoor field survey scenario. Within this test site, we plan to deploy the AROX robot autonomously in specified intervals to monitor the development of the crops

TABLE I
THE DIFFERENT ZONES, DOMAIN-SPECIFIC CONTROLLER AND RECOVERY CONFIGURATIONS, AND LOCALIZATION METHODS.

Zone	Type	Controller ID	Recovery ID	Localization
z1	**container**	docking	wait	AMCL
z2	**grassland**	dynamic_green	rotate	GNSS
z3	**field**	infield	wait	GNSS
z4	**field**	infield	wait	GNSS
z5	**grassland**	dynamic_green	rotate	GNSS
z6	**garden**	safety_ctrl	wait	GNSS
z7	**courtyard**	safety_ctrl	wait	GNSS
z8	**public road**	-	stop	GNSS

TABLE II
LIST OF CONTROLLER CONFIGURATIONS WITH PARAMETERS.

Controller ID	Controller Plugin	Basic Parameters
dynamic_green	EBand	vel_x: 0.1 - 3.3 m/s vel_rot: 0.05 - 0.8 rad/s xy_goal_tol: 1.0 m yaw_goal_tol: 0.8 rad
infield	EBand	vel_x: 0.1 - 2.2 m/s vel_rot: 0.05 - 0.4 rad/s xy_goal_tol: 0.15 m yaw_goal_tol: 0.25 rad
safety_ctrl	DWA	vel_x: 0.1 - 1.0 m/s vel_rot: 0.05 - 0.4 rad/s xy_goal_tol: 0.5 m yaw_goal_tol: 0.8 rad
docking	AI-trained custom	vel_x: 0.1 - 0.5 m/s vel_rot: 0.05 - 0.4 rad/s xy_goal_tol: 0.025 m yaw_goal_tol: 0.05 rad

on the surrounding fields. For maize, we aim to acquire a series of 3D laser scans with the terrestrial laser scanner every week at specific poses to document the growth of the crops. After the data acquisition, the collected data shall be sent to a central server where it is processed automatically. Up to now, we installed the container with the power charging station, network connection, and GNSS RTK reference base. With this setup, we are currently able to safely drive to specified target locations using different controllers and representations to take 3D laser scans. An exemplary point cloud created from multiple 3D laser scans is shown in Fig. 5. A virtual fly-through of that scene video demonstrating the high quality the recorded colored 3D-point is available at our Youtube channel[2].

For the task at hand, we manually created a map of the working area using digital ground models (DGM) provided by the National Agency for Geoinformation and State Survey of Lower Saxony (LGLN). Within this geo-referenced map, we marked the respective zones in UTM coordinates to allow localization of AROX using the installed differential GNSS system. On top of the zone descriptions, we added a topological waypoint graph to model the transitions between the different areas as shown in Fig. 2. Each zone in this map is associated with a certain controller, a list of recovery behaviors and localization techniques. The different settings and zones configurations in the presented environment are shown in Tab. I.

The *Controller ID* column in Tab. I and Tab. II correspond to each other and specify the callable controller name used in *ExePath*. Further, Tab. II shows the used controller / local planner plugin with the specific configuration. For our setup, we use the *elastic-band-planner* (EBand) [15], the dynamic window approach (DWA) [16], and a custom local planner to dock the robot to the charging station.

B. Navigation Example

In the scenario shown in Fig. 2, the robot is located in the *garden* zone *z6* and has to navigate to a specific point inside the northern *field* zone *z4* before returning via *z2* to the *container z1* for charging. As listed in Tab. I, the robot starts in the *garden* zone (z6) with an *safety_ctrl* controller. It uses the DWA local planner plugin with a configuration sketched in Tab. II to ensure an effective dynamic obstacles reaction.

When the robot passes the first waypoint into the *grassland* zone *z2*, the controller is switched to `dynamic_green`. To account for the available space and good driveability, the EBand controller plugin is configured with a higher maximum velocity and less strict pose tolerances. The last zone that the robot passes through in the direction of the goal position is the *field* zone *z4* using the *infield* controller. Currently this controller is based on a more restricted EBand parameterization as shown in Tab. II, with lower maximum speed and tolerances to the goal position, due to the expected high wheel slip in the loose soil.

To return to the container *z1* for battery charging, the robot has to traverse the *grassland* zone *z2* again as described before. The waypoint to enter the container is located just before the container's ramp.

By passing this point the system will have to switch from an outdoor multi-band RTK GNSS with fused odometry and IMU data as means of localization, to *AMCL* [4]. Currently, switching the corresponding ROS nodes have yet to be integrated to blend into the system as a whole. As further future steps we will integrate new path planners and controllers for the infield zone and additional maps for the outdoor scenario. Additionally, semantic annotation of the zones should be used not only to look up predefined configurations, but also to infer them using reasoning algorithms based on the well known *rete* algorithm [17]. Another step to accommodate for heterogeneous terrain navigation in addition to the 2D gridmap navigation, 3D polygonal representations will be provided for some areas, which can be used for 3D Mesh based navigation as described in previous work by Pütz et al. [1].

C. Towards long-term autonomy

With the current implementation, we realized a working prototype that is able to safely navigate within the modeled environment across different zones. To achieve self-sufficient long-term autonomy, besides robust navigation, an execution

[2] https://youtu.be/HyhkOWYah34

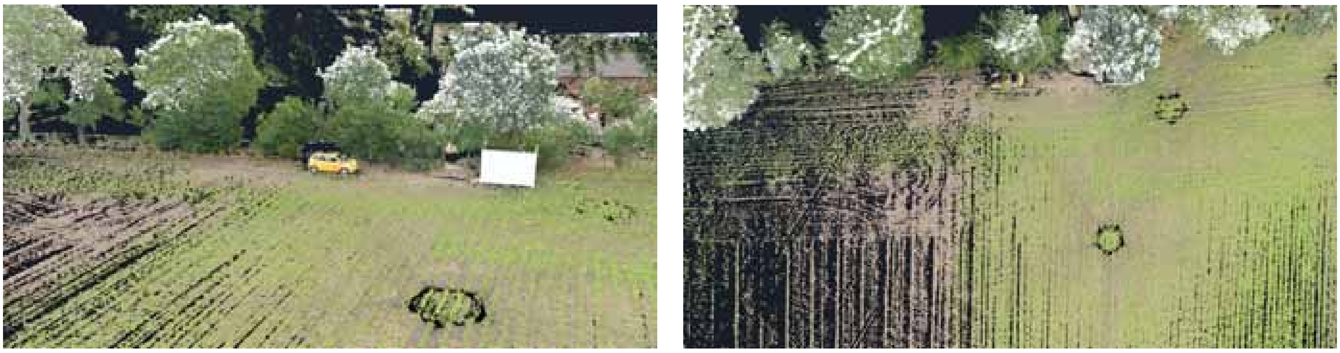


Fig. 5. Colored 3D point cloud of the data collected with the terrestrial laser scanner of the AROX robot. Color values are derived from the integrated RGB camery. Annotation with hyperspectral data is also possible.

monitoring system is required to detect unforeseen events and abnormal behavior of the system. For that, we plan to augment the current environment representation with a fully symbolic semantic model.

Having such a representation allows to use rule-based reasoning to detect process states. In previous work [18], we combined spatial representations like the one used in this scenario, to monitor machine states and generate process events in a maize harvesting campaign. For that, we used our SEMAP framework [11] that allows to deduct qualitative spatial relations with explicitly modeled semantic background knowledge about the involved machines, used facilities and process events.

The transfer of such a model into the currently developed system is straight-forward and would allow to monitor the current state of the autonomous surveying systems. In the given context, such a semantic monitoring system allows to automatically detect high level process states like ”charging”, ”driving” or ”scanning” based on the sensor data and semantic environment model. On the other hand, abnormal or illegal states could also be detected by defining simple rules like ”the AROX robot should never be outside the defined zones”, or ”the robot should never start a measurement campaign without having charged in the container before”. With an according spatio-semantic model, these and similar rules could be defined easily to improve the monitoring during the desired long-term deployment of the autonomous system.

VI. Discussion

In this paper we presented the first work towards a fully autonomous robotic system in an agricultural surveying scenario. The system is completely implemented in ROS allowing the use of proven standard methods as well as more application-specific algorithms and representations. The current state presented here serves as an extensible foundation to realize safe navigation in dynamic environments. In future work it will be combined with an actual semantic mapping framework to enable more complex planning and execution monitoring, and to increase the capabilities of the autonomous system.

References

[1] S. Pütz, T. Wiemann, J. Sprickerhof, and J. Hertzberg, “3d navigation mesh generation for path planning in uneven terrain,” in *9th IFAC Symposium on Intelligent Autonomous Vehicles (IAV 2016)*. IFAC, 2016.

[2] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, “The office marathon: Robust navigation in an indoor office environment,” in *ICRA 2010*. IEEE, 2010, pp. 300–307.

[3] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 116–121.

[4] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” 01 1999, pp. 343–349.

[5] S. Pütz, J. Simón, and J. Hertzberg, “Move base flex: A highly flexible navigation framework for mobile robots,” in *IROS 2018*, 2018.

[6] J. Bohren and S. Cousins, “The smach high-level executive [ros news],” *IEEE Robotics & Automation Magazine*, vol. 17, no. 4, pp. 18–20, 2010.

[7] D. C. Conner and J. Willis, “Flexible navigation: Finite state machine-based integrated navigation and control for ros enabled robots,” in *SoutheastCon 2017*, March 2017, pp. 1–8.

[8] D. V. Lu, D. Hershberger, and W. D. Smart, “Layered costmaps for context-sensitive navigation,” in *IROS 2014*. IEEE, 2014, pp. 709–715.

[9] B. Kuipers, “Spatial semantic hierarchy,” *Artificial Intelligence*, vol. 119, no. 1, pp. 191–233, 2000.

[10] M. Günther, J.-R. Ruiz-Sarmiento, C. Galindo, J. Gonzalez-Jimenez, and J. Hertzberg, “Context-aware 3d object anchoring for mobile robots,” *Robotics and Autonomous Systems (RAS)*, vol. 110, pp. 12–32, 12 2018.

[11] H. Deeken, T. Wiemann, and J. Hertzberg, “Grounding semantic maps in spatial databases,” *Robotics and Autonomous Systems*, vol. 105, pp. 146–165, 2018.

[12] A. Saffiotti, S. Coradeschi, P. Busch, and J. Gonza, “Multi-Hierarchical Semantic for Mobile Robotics,” in *IROS 2015*, 2005.

[13] P. Buschka, “An investigation of hybrid maps for mobile robots,” Ph.D. dissertation, Örebro universitetsbibliotek, 2005.

[14] D. Lang and D. Paulus, “Semantic Maps for Robotics,” *Robotics,Proc. of ICRA, ”Workshop on AI Robotics”*, 2014.

[15] S. Quinlan and O. Khatib, “Elastic bands: connecting path planning and control,” in *Proceedings IEEE International Conference on Robotics and Automation*, 1993, pp. 802–807 vol.2.

[16] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[17] C. L. Forgy, “Rete: A fast algorithm for the many pattern/many object pattern match problem,” in *Readings in Artificial Intelligence and Databases*. Elsevier, 1989, pp. 547–559.

[18] H. Deeken, T. Wiemann, and J. Hertzberg, “A spatio-semantic approach to reasoning about agricultural processes,” *Applied Intelligence*, vol. 2019, 2019.

Exploiting Continuity of Rewards: Efficient Sampling in POMDPs with Lipschitz Bandits

Ömer Şahin Taş, Felix Hauser, and Martin Lauer

***Abstract*— Decision making under uncertainty can be framed as a partially observable Markov decision process (POMDP). Finding exact solutions of POMDPs is generally computationally intractable, but the solution can be approximated by sampling-based approaches. These sampling-based POMDP solvers rely on multi-armed bandit (MAB) heuristics, which assume the outcomes of different actions to be uncorrelated. In some applications, like motion planning in continuous spaces, similar actions yield similar outcomes. In this paper, we utilize variants of MAB heuristics that make Lipschitz continuity assumptions on the outcomes of actions to improve the efficiency of sampling-based planning approaches. We demonstrate the effectiveness of this approach in the context of motion planning for automated driving.**

Index Terms— POMDP, multi-armed bandits, Monte Carlo planning, POSLB, POSLB-V, motion planning.

I. Introduction

Sequential decision making problems in which the system dynamics are uncertain and the system state is unobservable can be framed as a POMDP. The POMDP framework encodes uncertain and incomplete knowledge not by single states, but by beliefs over all possible states. By optimizing over a sequence of actions and observations, it considers a very large number of possible future outcomes. This comes at the cost of high computational complexity.

A POMDP can typically be solved either by performing *value iteration*, or by *Monte Carlo tree search*. The latter are real-time capable and are more flexible since they do not require state discretization. Common approaches are the algorithms POMCP [1] and TAPIR [2]. They employ the Upper Confidence Bound (UCB) [3] bandit algorithm to explore promising actions and exploit good ones.

UCB does not make any assumption on the outcomes of similar actions. This is essential for certain decision making problems, e.g. playing board games. However, many real-world applications operate on compact, continuous spaces in which the profile of outcomes are continuously differentiable. Furthermore, the uncertainties present in the environment have a *smoothing effect* on any discontinuity, if they can be represented with a probability distribution whose density is continuous. Such applications can benefit from Lipschitz continuity assumptions on the outcomes of actions.

A POMDP solver can exploit the dependence between the expected rewards of different actions by utilizing a MAB that assumes Lipschitz continuity. Even though this would substantially improve the efficiency of the sampling, currently there is no POMDP solver that exploits this property. In this paper, we propose to utilize a Lipschitz MAB that can work within the POMDP framework. We further investigate whether the motion planning problem in automated driving has a continuous reward profile by analyzing the dependencies between actions in different scenarios, including edge cases like collisions.

In order to highlight the contribution of this work, we first provide background information on POMDPs in Section II. Next, in Section III we focus on existing works that deal with motion planning for automated vehicles by utilizing the POMDP framework and subsequently introduce our model, which shows several differences to existing works. Once the underlying settings are introduced, we provide an overview on MABs we benchmark in Section IV. We use a modified version of the POMCP algorithm to efficiently solve the POMDP problem, presenting those modifications in Section V, before analyzing the structure of reward profiles and benchmarking the MABs in the Evaluation section.

II. Background

A POMDP operates on spaces of states $\mathbb{S}$, observations $\mathbb{O}$ and actions $\mathbb{A}$. The state s cannot be observed exactly, and therefore, it is represented as a probability distribution over the state space, i.e. belief $b(s)$. Transitions from one state to another and their respective observations are given by a transition and an observation model. Every time the agent chooses an action $a \in \mathbb{A}$ and the environment transitions to the next state, the agent receives the scalar reward $r \in \mathbb{R}$, computed by the reward function $\mathcal{R}(s, a)$. The action-value $Q^{\pi}(b, a)$ represents the expected future reward when taking action a in belief b and following policy π afterwards.

The POMCP algorithm [1] applies Upper Confidence Bound for Trees (UCT) [4] to POMDPs by iteratively sampling sequences of states and observations. These sequences are kept track of in a tree data structure of nodes and edges, corresponding to states, actions and observations. Key to the POMCP algorithm is the equivalence of belief b and history h. A history is the sequence of actions and observations that have been selected and observed by the agent, $h_t = \{a_1, o_1, \dots, a_t, o_t\}$. When the initial belief b_0 is fixed and known, the belief is represented by the history.

One episode of the sampling procedure samples a single particle and involves four phases: In the `simulation` phase of POMCP, actions are selected by the MAB algorithm. Based on the generative observation model, the next state is determined. When simulation reaches a node that is not

Corresponding author: `tas@fzi.de`. The authors are with FZI Research Center for Information Technology and Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany.

in the tree yet, the tree is `expanded` by the node and the simulation stops. The initial value estimate of the new node is done by `rollout` policy, which is used as an heuristic in lieu of expanding the tree further. In the final phase, `backup`, the encountered nodes are updated from bottom-up in light of the newly gathered information.

III. Motion Planning in Automated Driving with the POMDP Framework

In automated driving, motion planning has to consider uncertain information such as the unknown intentions and future directions of other traffic participants, or objects in occluded areas. One way to consider these uncertainties is to use the POMDP framework.

A. Related Work

POMDPs have been used in previous works to solve the motion planning problem for automated driving. Bai et al. [5] use the POMCP algorithm for planning in an intersection scenario. They model the uncertain intentions of other drivers and driving style. Although the POMCP solver is capable of handling continuous state spaces, the authors used discrete states. Brechtel introduces a POMDP solver that is able to work with continuous spaces by learning a problem-specific representation of the state space [6], [7].

The importance of considering uncertainties in the planning of motion for automated driving is shown by Sunberg et al. [8]. They compare different planning frameworks based on MDPs and POMDPs, which are evaluated on a lane change scenario. The results clearly indicate the need to take uncertainties into consideration. For the POMDP variant the POMCP algorithm is enhanced with a technique called double progressive widening [9].

Sefati et al. [10] also include uncertainties in the motion planning for an intersection scenario. They consider the unknown intentions of other traffic participants by inferring the state with a Bayesian network model. To solve the POMDP, the MCVI solver [11] is combined with ideas form the SARSOP algorithm [12]. Their approach uses several heuristics to guide the exploration through the action space, though sparse, containing only three actions.

Bouton et al. [13] study intersection and pedestrian crosswalk scenarios. They do not deal with intentions of other road users, but focus on the integration of occlusions. The action space includes only four or five actions, depending on the scenario. The POMDP problem with multiple participants is split into many problems involving only one other agent.

Hubmann et al. [14] use a real-time method for computing solutions to intersection scenarios with multiple vehicles with unclear intentions. They use the TAPIR algorithm and consider an action space of only four actions. In a later work they extend their approach to tackle occlusions [15].

B. Modeling the Motion Planning Problem for the POMDP Framework

The POMDP framework requires model-based representations on which to operate. We closely follow the model presented in [14] with some modifications to work better with a denser action space.

1) Map data: We refer to a path and its accompanying data as a *route* and denote it by ρ. Accompanying data consists of the curvature κ and the reference velocity v. In this way, we store every route ρ as a tuple of n points

$$\rho = (p_i)_{i=1,\ldots,n} \quad \text{with} \quad p_i = (x_i, y_i, l_i, \kappa_i, v_i)^\top \in \mathbb{R}^5,$$

where x and y are Cartesian coordinates and l is the distance along the path.

2) States, Observations, Actions: We apply path-velocity decomposition [16] to reduce the complexity and, therefore, we plan acceleration along a path. The state s of a vehicle k is given by $s_k = (l_k, v_k, \rho_k)$. For the ego vehicle the route ρ is known, hence, we represent its state as $s_0 = (l_0, v_0)$. For k other vehicles in a traffic scene, we have the combined state $s = (s_0, s_1, s_2, \ldots, s_k)$.

Observations o are defined in Cartesian coordinates and only contain information about k other traffic participants $o = (o_1, o_2, \ldots, o_k)$ with $o_k = (x_k, y_k, v_k)^\top \in \mathbb{R}^3$.

Actions available to the planner represent different values of acceleration a. Throughout the rest of this work the available acceleration values are within the comfortable range $a \in [-3\,\mathrm{m\,s^{-2}}, 1\,\mathrm{m\,s^{-2}}]$ with equidistant spacing.

3) Transition model: The POMDP model is discretized in time and the transitions have the timestep $T_s = 1.0$s. The vehicles follow a constant acceleration model and the route does not change. For the ego vehicle the route is predefined and the acceleration input a_0 is simply the action chosen by the POMDP solver.

For predicting accelerations of other vehicles a_k, we use the Intelligent Driver Model (IDM) [17]. This model consists of a free driving term $a_{\text{ref},k}$ that allows the vehicle to smoothly adjust the velocity in case of deviations from its reference, and an interaction term $a_{\text{int},k}$ for avoiding collisions. We saturate the resulting acceleration value with the maximum deceleration a^{-} and add Gaussian noise $a_{\text{noise},k} \sim \mathcal{N}(0, \sigma_a^2)$ to cover for modeling errors and uncertainties in the behavior model.

$$a_k = \max(a_{\text{ref},k} + a_{\text{int},k},\ a^{-}) + a_{\text{noise},k}.$$

IDM may yield accelerations resulting in negative velocity due to time discretization. In such cases, we calculate the stop position from the motion kinematics.

4) Observation model: The observation model generates a possible observation from a given state-particle. After the transformation from path coordinates to Cartesian coordinates, we add observation noise sampled from independent Gaussian distributions.

5) Reward model: The terms in the reward model resemble those of an ordinary trajectory optimization problem. All terms are modeled as negative rewards

$$r = r_{\text{coll}} + r_v + r_{j,\text{lon}} + r_{a,\text{lat}}.$$

The collision term r_{coll} takes only collisions of the ego vehicle into account

$$r_{\text{coll}} = \begin{cases} 0 & \text{no collision} \\ \zeta_{\text{coll}} & \text{ego vehicle collides} \end{cases} .$$

The criterion from [18, p. 223] is applied for the collision check, which is performed during state transition. The corresponding reward therefore is a function of the current state and the previous one.

We punish the deviation of the ego vehicle velocity from a predefined reference with an asymmetric loss function

$$r_v = \begin{cases} \zeta_v \ (v_0 - v_{\text{ref}})^2 & \text{if } v_0 \geq v_{\text{ref}} \\ \zeta_v \ \log\Big(1 + (v_0 - v_{\text{ref}})^2\Big) & \text{otherwise} \end{cases} ,$$

with ζ_v being the cost scaling factor. We apply quadratic cost if v_{ref} is exceeded and *Cauchy loss* [19] otherwise. The asymmetric loss function is motivated by the assumption, that driving slow or standing still might be part of an ordinary solution to the motion planning problem, whereas driving with higher speed is only acceptable in extraordinary cases.

The third term in the reward model $r_{j,\text{lon}} = \zeta_{j,\text{lon}} {j_0}^2$ accounts for jerk and considers changes in the longitudinal acceleration. The last term is the lateral acceleration term $r_{a,\text{lat}} = \zeta_{a,\text{lat}} \left(\kappa \ v_0^2\right)^2$.

IV. Multi-armed bandits

The general MAB is a sequential decision problem [20], which proceeds in rounds denoted by $t \in \{1, \ldots, T\}$. At every round, the agent picks one arm a from some set of arms $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$, where K denotes the number of arms, with the goal of maximizing the rewards it collects over T rounds. In the stochastic setting, every arm corresponds to a reward distribution, whose properties are unknown to the agent. The distributions are assumed to be stationary, i.e. they do not change over time. The MAB algorithm balances between the exploration of unknown arms and exploitation of good arms.

Several approaches have been suggested in the past to deal with the MAB problem.

A. UCB

Introduced by Auer et al. [3], UCB is *optimistic in the face of uncertainty*. Its robustness and practicality have been proven in numerous applications. In each round t the algorithm calculates the index

$$b_t(a) = \hat{\mu}_t(a) + c\sqrt{\frac{2\log t}{n_t(a)}} \tag{1}$$

for every arm and then chooses the arm with the highest index (cf. Algorithm 1). The first term is the current average reward of an arm $\hat{\mu}_t(a)$. The second term is called the confidence radius and is proportional to the upper bound of the confidence interval of the average reward. The denominator $n_t(a)$ is the number of times arm a has been played and $c \in \mathbb{R}$ is the *exploration constant* trading-off exploitation and exploration.

Algorithm 1: Upper Confidence Bound (UCB)

```
if $t \leq K$ then
    Choose arm from $\{a : n_t(a) = 0\}$ at random
else
    Choose arm $a_t = \arg\max_{a \in \mathcal{A}} b_t(a)$
```

A more general case of UCB is the KL-UCB bandit algorithm. It allows distributions of canonical exponential family to be set for the reward. If rewards of the arms are assumed to be Gaussian distributed with equal variance, the KL-UCB index becomes equal to Equation 1, but with an exploration constant of $c = \sqrt{\sigma^2}$.

B. UCB-V

Audibert et al. enhanced UCB by including the estimated reward variances $\hat{\sigma}_t^2(a)$ [21]. In contrast to UCB, the variances of the rewards are not assumed to be equal. The index is given as

$$b_t(a) = \hat{\mu}_t(a) + \sqrt{\frac{2\hat{\sigma}_t^2(a)\log t}{n_t(a)}} + \frac{3c\log t}{n_t(a)}. \tag{2}$$

UCB-V still has the exploration constant in the third term, but with growing $n_t(a)$ the exploration constant loses influence and the estimated variances are more strongly considered. To be precise, the second term has $\sqrt{n_t(a)}$ times the weight of the third term.

C. POSLB

The Pareto Optimal Sampling for Lipschitz Bandits (POSLB) algorithm assumes the expected rewards to be Lipschitz continuous and thereby improves the efficiency of sampling by guiding the bandit faster to more rewarding arms [22, p. 21]. Although Lipschitz continuity in the reward function values is assumed, the set of arms is still discrete. The expected reward function over arms is assumed to obey

$$|\mu(a) - \mu(a')| \leq \mathcal{L}\,|a - a'|$$

for any pair of arms (a, a') and a Lipschitz constant $\mathcal{L}$. The POSLB algorithm for Kullback-Leibler divergence of Gaussian distributed rewards is given in Algorithm 2. The algorithm identifies the currently best arm a_t^* and calculates the KL-UCB index $b_t(a_t^*)$. The intermediate values $\big(\lambda_t(a, a_1), \ldots, \lambda_t(a, a_K)\big)$ can be understood as the *most confusing* estimated reward vector, which would make the suboptimal arm a the optimal one. The Lipschitz assumption is integrated into the bandit via the confusing rewards and is adjusted by $\mathcal{L}$. POSLB looks at the differences between the most confusing rewards and the actual estimated rewards and favors arms where the sum of the differences, weighted by their visit counts, is small.

The algorithm runs with an increased complexity of $O(|\mathcal{A}|^2)$ compared to UCB.

Algorithm 2: POSLB

```
if $t \leq K$ then
    Choose arm from $\{a : n_t(a) = 0\}$ at random
else
    $a_t^* = \arg\max_{a\in\mathcal{A}} \hat{\mu}_t(a)$
    $f_t(a) = \begin{cases} \sum_{a'\in\mathcal{A}} n_t(a')\big(\hat{\mu}_t(a') - \lambda_t(a,a')\big)^2 (2\sigma^2)^{-1} & \text{if } a \neq a_t^* \\ n_t(a)\big(\hat{\mu}_t(a) - b_t(a_t^*)\big)^2 (2\sigma^2)^{-1} & \text{if } a = a_t^* \end{cases}$
    with
      $\lambda_t(a,a') = \max\big(b_t(a_t^*) - \mathcal{L}\,|a-a'|\,,\ \hat{\mu}_t(a')\big)$
    Choose arm $a_t = \arg\max_{a\in\mathcal{A}} \log t - f_t(a)$
```

D. POSLB-V

While POSLB is able to consider the Lipschitz continuity of the reward function, it does not make use of estimated variances. Simply replacing the variance parameter σ^2 in the POSLB algorithm by the estimated variances $\hat{\sigma}^2(a)$ leads to poor results. Instead, it is advisable to shift from exploration constant to estimated variances over time, like it is done in UCB-V.

Equating the KL-UCB and UCB-V indices and solving for the variance parameter leads to

$$\sigma_t^2(a) = \sigma^2 = \frac{n_t(a)}{2\log t}\left(\sqrt{\frac{2\hat{\sigma}_t^2(a)\log t}{n_t(a)}} + \frac{3c\log t}{n_t(a)}\right)^2 .$$

We then use the estimated variances and Lipschitz assumption together and and call it POSLB-V.

The augmented variance $\sigma_t^2(a)$ is then used in POSLB to calculate $b_t(a_t^*)$ and $f_t(a)$, instead of the fixed σ^2. Otherwise, the algorithm stays unchanged.

E. Continuous bandits

All bandits above need to discretize the continuous action space. However, there are several bandit algorithms that can handle continuous action spaces [23]–[25]. Other MABs additionally assume that the returns of the arms are Lipschitz continuous [26]–[29]. An overview is provided in [20, p. 40]. In every round they choose a new arm from the action space, never sampling the same action twice. This prohibits the usage of these bandits within the POMCP algorithm, which needs to build a belief tree. The bandit in [30] can deal with a large amount of discrete actions, but cannot easily be utilized in a belief tree. None of these approaches is compatible with real-time POMCP and, therefore, are not investigated further.

V. Experimental Settings

We adapt the POMCP algorithm [1] by modifying the `simulation`, `rollout`, and `backup` phases.

In the `simulation` phase of POMCP, we compare the MABs presented in the previous sections for choosing actions.

In the `rollout` phase we perform constant velocity rollouts for their minimal computational time.

We modify the `backup` phase in several ways. We use incremental statistics [31] to calculate the mean and the variance of Q-values

$$Q_n = Q_{n-1} + \eta(n)\ \big(\widehat{Q} - Q_{n-1}\big), \tag{3}$$

$$\sigma_n^2 = \big(1 - \eta(n)\ \big)\Big(\sigma_{n-1}^2 + \eta(n)\big(\widehat{Q} - Q_{n-1}\big)^2\Big). \tag{4}$$

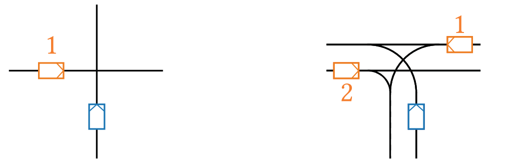
We alter the learning rate $\eta(n) = \frac{1}{n^{\omega}}$ by choosing a *polynomial learning rate* $\omega < 1$ instead of a *linear* rate where $\omega = 1$. As pointed out in [32], setting $\omega = 0.77$ has superior convergence properties. We use the maximum of Q-values as an estimate of the belief nodes value

$$V(h) = \max_{a\in\mathcal{A}} Q(h,a).$$

The belief tree of the POMDP solver works only with discrete observations. Therefore, we discretize the continuous observation space in a data stream clustering manner: A list of cluster centers is maintained and every new observation is compared to the entries in this list. If the Euclidean distance between observation and a cluster center is within a given threshold, the observation is assigned to the first matching cluster encountered in the ordered list. Otherwise, a new cluster center is inserted at the end of the list.

VI. Evaluation

We used two simple traffic scenarios "straight driving" and "traversing curves" for initial testing and development. For evaluation, we use two complex traffic scenes in which interaction with other participants are required (cf. Fig. 1). As the solution depends on the initial belief, we sample the state-particles of the initial belief from the same probability distribution. We assume that the distributions of position, velocity and route are independent.



(a) Collision scene. (b) Intersection scene.

Fig. 1: Traffic scenes used in the evaluations.

Collisions pose a counter-example to the underlying assumption in this work: they introduce an abrupt change in the reward and hence pose the most challenging problem. In the collision scenario $\texttt{S}_{\texttt{Coll}}$ we simulate the case of an imminent collision. In the intersection scene, the ego vehicle has to identify whether the other vehicles are on collision paths and avoid collisions. We define two scenarios based on this scene: the $\texttt{S}_{\texttt{I-Lo}}$ scenario and the $\texttt{S}_{\texttt{I-Hi}}$ scenario, which pose low and high probabilities of collision, respectively. The parameters of the scenarios are provided in Appendix.

A. Q-value Function Analysis

If the transition model and reward function of the ego vehicle are Lipschitz continuous, the resulting Q-value function of an MDP is guaranteed to be Lipschitz continuous as well [33]. To expand the result to POMDPs, intuitively the observation function needs to be Lipschitz continuous as well. This is not the case in our model, as the reward includes the binary collision term, rendering the Q-value function discontinuous. We argue, however, that the noise, which is present in the transition and observation model, has a smoothing effect on these discontinuities.

In order to empirically analyze the continuity of Q-value function, we accurately evaluate it in the root node of the belief tree. We set up equidistant actions with $\Delta a = 0.05\,\mathrm{m\,s^{-2}}$ and thereby cover the action space densely. Then, the Q-values of these actions are evaluated by conducting a simulation run for each of the actions with 10^6 particles. During a single run, the action in the root node is kept fixed, whereas in the following belief nodes UCB is used to select among five available actions.

To highlight the discontinuity in the reward, we evaluate $\texttt{S}_{\texttt{Coll}}$ without uncertainties (cf. Fig. 2). If the agent chooses an action between $-2.0\,\mathrm{m\,s^{-2}}$ and $-0.2\,\mathrm{m\,s^{-2}}$ a collision cannot be prevented. The step-like patterns emerge due to discretized values of braking actions in subsequent timesteps.

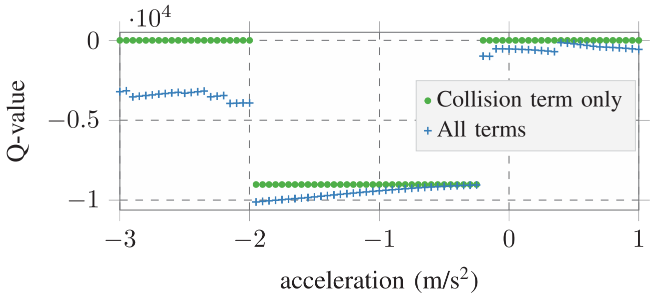


Fig. 2: Q-value profile of $\texttt{S}_{\texttt{Coll}}$ without uncertainty.

Coarse and more accurate approximations of the Q-value profiles of $\texttt{S}_{\texttt{Coll}}$ are presented in Fig. 3. The coarse one is approximated with 10^4 particles, whereas the more accurate one is obtained by sampling 10^7 particles with 17 actions, and doubled resolution of the observation discretization. The Q-value profile converges to a continuous function. From the more accurate simulation, it can be seen that the variance is reduced. However, as a result of reduced smoothing, the variance between successive actions tend to be higher than the rest at discontinuities. This points out that the Lipschitz assumption loses its validity for huge amount of samples under low uncertainty.

The Q-value profile of $\texttt{S}_{\texttt{I-Lo}}$ and $\texttt{S}_{\texttt{I-Hi}}$ are presented in Fig. 4. Both are very smooth and have the same underlying shape, whereas the overall level of Q-values is less and the variances are higher in $\texttt{S}_{\texttt{I-Hi}}$, as a result of the higher collision probability.

The continuity evaluation illustrates that the uncertainties partially smooth the discontinuities. Increasing the number of available actions increases the smoothness of the profile, as well. Notice that we model uncertainties in the longitudinal direction only. Considering the lateral position uncertainty would further smooth the Q-values.

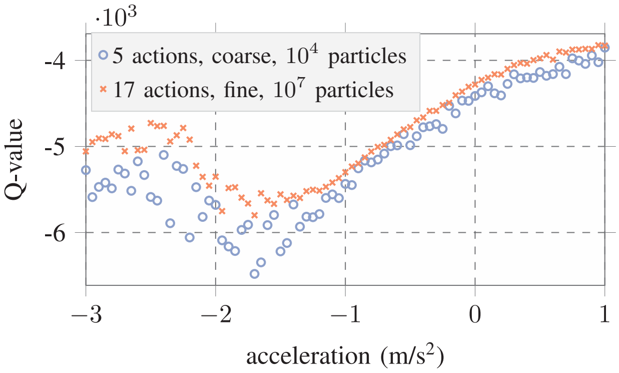


Fig. 3: Approximations of the Q-value profile of $\texttt{S}_{\texttt{Coll}}$.

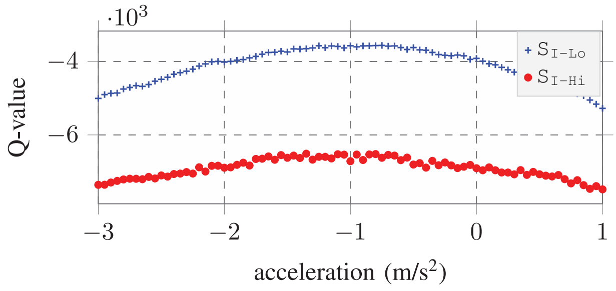


Fig. 4: Approximated Q-value profiles of the intersection scenarios sampled with 10^4 particles.

B. Convergence Analysis

A standard metric to benchmark convergence is to identify the optimal action after a predefined number of samples n. We use the *mean absolute error* (MAE) between the current best action a_n^* and the optimal action a^* as a performance measure.

We calculate a^* by sampling 10^7 particles while employing UCB bandit. The observed Q-values still have stochastic nature and hence, we perform Gaussian process regression to eliminate the noise of the Q-values and to recover the underlying function. As the profile of the Q-values is sufficiently smooth, we use a squared exponential kernel with length scale and noise level as hyperparameters. We estimate the optimal action for every scenario and number of available actions by evaluating the mean of the fitted Gaussian process at the locations of the available actions. The results are given in TABLE I in APPENDIX.

To perform a reliable analysis, we calculate the MAE over multiple simulation runs m

$$\mathrm{MAE}_n = \frac{1}{m} \sum_{i=0}^{m-1} \left| a_{i,n}^* - a^* \right| ,$$

where $m = 100$ and $2 \cdot 10^4$ episodes. Given the ground truth, we determine a_n^* as

$$a_n^* = \arg\max_{\mathrm{a} \in \mathcal{A}} \; Q_n(h_0, a).$$

The optimal Lipschitz constant required for POSLB is not known in advance. Overestimating and underestimating lead to suboptimal performance. We use the mean of the fitted Gaussian process and empirically select the Lipschitz constant $\mathcal{L} = 2000$ for these scenarios (cf. TABLE II in APPENDIX).

Fig. 5 presents the convergence results for $\texttt{S}_{\texttt{Coll}}$. From the figures, it is clear that for a low number of available actions all of the bandits have comparable convergence properties. However, as the number of actions increases, POSLB and POSLB-V show superior performance. In the case of 33 actions, the UCB bandits have twice the MAE compared to Lipschitz bandits after $2 \cdot 10^4$ episodes. The variants considering variances perform comparably. Another obvious result is that none of them can reach zero MAE. The value they reach is equal to the action discretization Δa, as expected. Even though not presented, the results for 17 actions lay between those for 9 and 33.

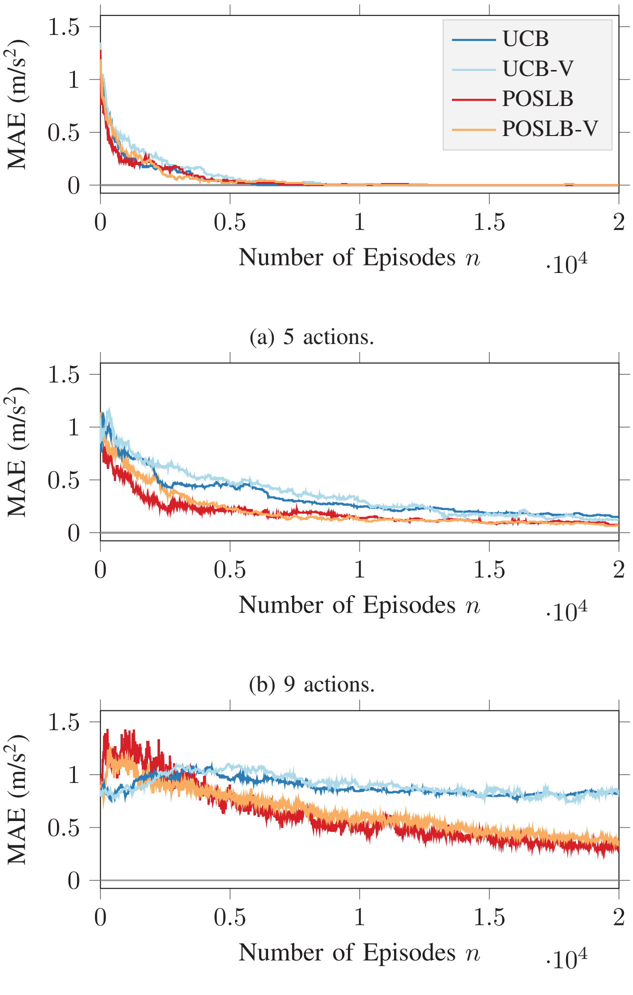
The results for 9 and 33 actions of $\texttt{S}_{\texttt{I-Lo}}$ are given in Figure 6. The results for $\texttt{S}_{\texttt{I-Hi}}$ are very similar to those of $\texttt{S}_{\texttt{Coll}}$ (cf. Fig. 5), as expected. Strikingly, the POSLB bandit shows the slowest convergence in the case of 33 actions, and POSLB-V performs best. The poor performance of POSLB is caused by misleading rollouts which point to a different area of the action space to be optimal. The Lipschitz assumption causes the bandit to select actions in that area. By selecting actions with higher variances more often, POSLB-V compensates the drawbacks resulting from such misleading rollouts. $\texttt{S}_{\texttt{I-Lo}}$ resembles such a narrow case in which the consideration of variances is advantageous.

C. Discussion

The results of the convergence analysis present the average over $m = 100$ runs. We analyze the standard deviation of MAE (σ_{MAE}) for individual runs of different bandits. The results indicate that the σ_{MAE} values are comparable, whereas POSLB bandits have slightly smaller σ_{MAE} then their UCB counterparts. The results for 9 actions in $\texttt{S}_{\texttt{Coll}}$ are presented in TABLE III in APPENDIX as an arbitrary example.

Tree depth of a solution is an important indicator of the quality: deeper trees consider longer horizons and are more accurate. In Fig. 7 we compare the tree depth for UCB and POSLB bandits for $\texttt{S}_{\texttt{Coll}}$ with the same number of particles. The bars in the figure represent the number of created nodes, and the color scales represent the number of visits for each level of the tree. It is clear that UCB has a greater branching factor compared to POSLB, resulting in shallower trees. In our experiments, other scenarios have verified this result.

Although the Lipschitz bandits have an increased computational complexity over UCB, we observed in our experiments that their runtime is never longer than 10% longer. This applies to the most demanding case of 33 actions. On average, POSLB is 5% slower.



(a) 5 actions.

(b) 9 actions.

(c) 33 actions.

Fig. 5: Mean absolute error for $\texttt{S}_{\texttt{Coll}}$, for different numbers of actions.

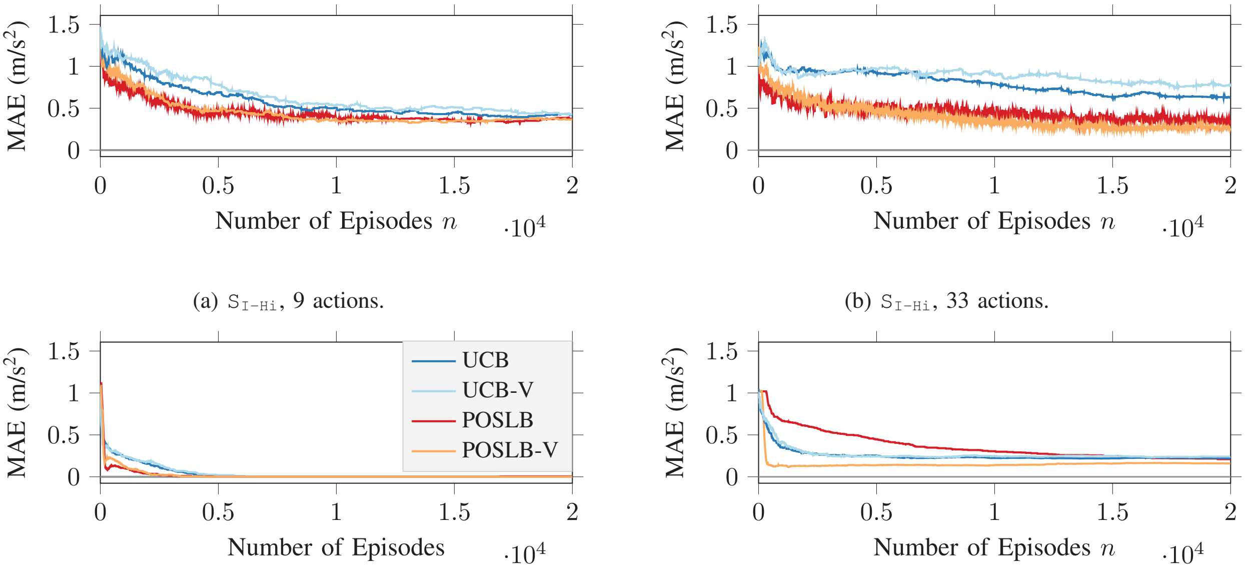
VII. CONCLUSIONS

In this paper, we present two important results. As a first result, we empirically show that the uncertainty in the transition and observation models of the POMDP formulation have a smoothing effect on the discontinuities in the Q-value function, eventually allowing for a Lipschitz continuity assumption.

We further show that the planning problem can be solved with fewer samples by utilizing the continuity of Q-values. By replacing the standard multi-armed bandit (UCB) with one that assumes Lipschitz continuity (POSLB), considerable performance improvements are achieved for higher numbers of actions, especially in the early stages of sampling.

A further contribution is the POSLB-V bandit that is derived from the POSLB bandit. Motivated from UCB-V, it considers the variance of Q-values during the action selection. Experiments have shown that considering variances can be advantageous, in cases where the rollout policy might be misleading.

Real-time capability constraints have limited the use of



(a) $\texttt{S}_{\texttt{I-Hi}}$, 9 actions.

(b) $\texttt{S}_{\texttt{I-Hi}}$, 33 actions.

(c) $\texttt{S}_{\texttt{I-Lo}}$, 9 actions.

(d) $\texttt{S}_{\texttt{I-Lo}}$, 33 actions.

Fig. 6: Mean absolute error for the intersection scenarios for different number of actions.

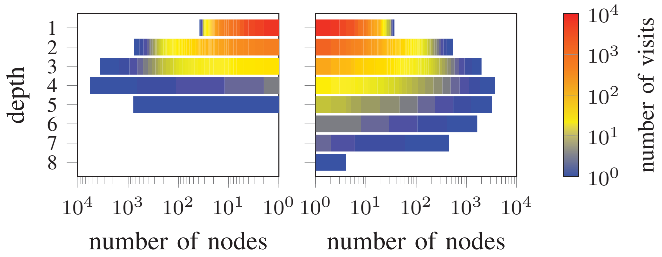


Fig. 7: Tree depth of UCB (left) and POSLB (right) bandits in $\texttt{S}_{\texttt{Coll}}$ for the same number of episodes.

existing POMDPs to decision making problems with fewer actions. With this work, we are able to accelerate the speed of POMDPs with a novel tree expansion technique that exploits the Q-value structure of our problem. This enables the use of POMDPs for problems where multiple actions need to be considered, such as in motion planning.

Appendix

TABLE I: Optimal action a^* $(\mathrm{m\,s^{-2}})$.

$\lvert\mathcal{A}\rvert$	Straight	Curve	$\texttt{S}_{\texttt{Coll}}$	$\texttt{S}_{\texttt{I-Lo}}$	$\texttt{S}_{\texttt{I-Hi}}$
5	0.0	−1.0	1.0	−1.0	−1.0
9	0.0	−1.5	1.0	−1.0	−1.5
17	0.0	−1.5	1.0	−1.0	−1.25
33	0.0	−1.0	0.875	−0.875	−1.125

TABLE II: Estimated Lipschitz constant $\mathcal{L}$ $(\mathrm{s^2\,m^{-1}})$.

$\lvert\mathcal{A}\rvert$	Straight	Curve	$\texttt{S}_{\texttt{Coll}}$	$\texttt{S}_{\texttt{I-Lo}}$	$\texttt{S}_{\texttt{I-Hi}}$
5	1247	1847	1003	1241	573
9	1271	2157	1981	1345	728
17	1336	2280	2742	1453	787
33	1370	2246	1260	1432	1033

Bandit	Number of Episodes						
	10^0	10^1	10^2	10^3	10^4	$2 \cdot 10^4$	10^5
UCB	0.57	0.57	0.81	0.55	0.29	0.23	0.07
UCB-V	0.52	0.52	0.66	0.53	0.32	0.21	0.09
POSLB	0.49	0.49	0.90	0.55	0.25	0.18	0.00
POSLB-V	0.65	0.65	0.77	0.57	0.21	0.17	0.07

TABLE III: σ_{MAE} for 9 actions of $\texttt{S}_{\texttt{Coll}}$.

TABLE IV: Times until the point-of-conflicts with different routes for the vehicles presented in the scenarios.

Scenario	Vehicle	Time-to-Intersection (s)			
$\texttt{S}_{\texttt{Coll}}$	ego	2.11			
	vehicle2	2.71			
$\texttt{S}_{\texttt{I-Lo}}$	ego	5.33	5.14	6.81	
	vehicle1	3.99	4.20	4.78	
	vehicle2	6.35	6.14	7.89	7.73
$\texttt{S}_{\texttt{I-Hi}}$	ego	2.66	2.28	5.63	
	vehicle1	2.78	3.23	4.52	
	vehicle2	3.42	3.05	6.21	5.92

TABLE V: Parameters used for evaluation.

Parameter	Value	Unit
ζ_{ref}	1	-
ζ_{lon}	35	-
ζ_{lat}	50	-
T_s	1	s
a^-	-3	$\mathrm{m\,s^{-2}}$
σ_a	3	$\mathrm{m\,s^{-2}}$
$t_{c,\min}$	1	s
$t_{c,\max}$	5	s
l_{goal}	15	m
l_{veh}	2	m
l_{idm}	2	m
T_{idm}	1.5	s
a_{cmf}	0.73	$\mathrm{m\,s^{-2}}$
b_{cmf}	1.67	$\mathrm{m\,s^{-2}}$
$\sigma_{o,\text{pos}}$	0.2	m
$\sigma_{o,\text{vel}}$	1.0	$\mathrm{m\,s^{-1}}$
d_{thresh}	1	m
ζ_{coll}	$-10\,000$	-
ζ_v	-100	-
$\zeta_{j,\text{lon}}$	-100	-
$\zeta_{a,\text{lat}}$	-100	-
ω	0.77	-
γ	0.95	-
c	10 000	-
$\mathcal{L}$	2000	$\mathrm{s^2\,m^{-1}}$
$d_{\text{roll},\max}$	20	-

REFERENCES

[1] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Advances in Neural Information Processing Systems*, 2010, pp. 2164–2172.

[2] D. Klimenko, J. Song, and H. Kurniawati, "Tapir: A Software Toolkit for Approximating and Adapting Pomdp Solutions Online," in *Proceedings of the Australasian Conference on Robotics and Automation, Melbourne, Australia*, vol. 24, 2014.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, May 2002.

[4] L. Kocsis and C. Szepesvári, "Bandit Based Monte-Carlo Planning," in *Machine Learning: ECML 2006*, ser. Lecture Notes in Computer Science, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Springer Berlin Heidelberg, 2006, pp. 282–293.

[5] Y. Bai, Z. J. Chong, M. H. Ang, and X. Gao, "An online approach for intersection navigation of autonomous vehicle," in *IEEE International Conference on Robotics and Biomimetics*, 2014, pp. 2127–2132.

[6] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *International IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 392–399.

[7] S. Brechtel, "Dynamic Decision-making in Continuous Partially Observable Domains: A Novel Method and its Application for Autonomous Driving." Ph.D. dissertation, Karlsruhe Institute of Technology, 2015.

[8] Z. N. Sunberg, C. J. Ho, and M. J. Kochenderfer, "The value of inferring the internal state of traffic participants for autonomous freeway driving," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 3004–3010.

[9] M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for autonomously navigating urban intersections," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 825–830.

[10] M. Sefati, J. Chandiramani, K. Kreisköther, A. Kampker, and S. Baldi, "Towards tactical behaviour planning under uncertainties for automated vehicles in urban scenarios," in *IEEE International Conference on Intelligent Transportation Systems*, 2017, pp. 1–7.

[11] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, "Monte Carlo value iteration for continuous-state POMDPs," in *Algorithmic Foundations of Robotics IX*. Springer, 2010, pp. 175–191.

[12] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces." in *Robotics: Science and Systems*, 2008.

[13] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Scalable Decision Making with Sensor Occlusions for Autonomous Driving," in *IEEE International Conference on Robotics and Automation*, May 2018, pp. 2076–2081.

[14] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 5–17, Mar. 2018.

[15] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller, "A pomdp maneuver planner for occlusions in urban scenarios," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 2172–2179.

[16] K. Kant and S. W. Zucker, "Toward Efficient Trajectory Planning: The Path-Velocity Decomposition:," *The International Journal of Robotics Research*, Jul. 2016.

[17] M. Treiber, A. Hennecke, and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000.

[18] C. Ericson, *Real-Time Collision Detection*. CRC Press, 2004.

[19] J. T. Barron, "A General and Adaptive Robust Loss Function," *arXiv:1701.03077 [cs, stat]*, Jan. 2017.

[20] A. Slivkins, "Introduction to multi-armed bandits," *arXiv preprint arXiv:1904.07272*, 2019.

[21] J.-Y. Audibert, R. Munos, and C. Szepesvári, "Tuning Bandit Algorithms in Stochastic Environments," in *Algorithmic Learning Theory*, ser. Lecture Notes in Computer Science, M. Hutter, R. A. Servedio, and E. Takimoto, Eds. Springer Berlin Heidelberg, 2007, pp. 150–165.

[22] S. Magureanu, "Efficient Online Learning under Bandit Feedback," Ph.D. dissertation, KTH Royal Institute of Technology, 2018.

[23] C. Mansley, A. Weinstein, and M. Littman, "Sample-based planning for continuous action markov decision processes," in *Twenty-First International Conference on Automated Planning and Scheduling*, 2011.

[24] A. Weinstein and M. L. Littman, "Bandit-based planning and learning in continuous-action markov decision processes," in *International Conference on Automated Planning and Scheduling*, 2012.

[25] A. D. Bull *et al.*, "Adaptive-treed bandits," *Bernoulli*, vol. 21, no. 4, pp. 2289–2307, 2015.

[26] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM, 2008, pp. 681–690.

[27] S. Bubeck, G. Stoltz, C. Szepesvári, and R. Munos, "Online optimization in x-armed bandits," in *Advances in Neural Information Processing Systems*, 2009, pp. 201–208.

[28] O.-A. Maillard and R. Munos, "Online learning in adversarial lipschitz environments," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 305–320.

[29] S. Bubeck, G. Stoltz, and J. Y. Yu, "Lipschitz bandits without the lipschitz constant," in *International Conference on Algorithmic Learning Theory*. Springer, 2011, pp. 144–158.

[30] E. Wang, H. Kurniawati, and D. P. Kroese, "CEMAB: A cross-entropy-based method for large-scale multi-armed bandits," in *Australasian Conference on Artificial Life and Computational Intelligence*. Springer, 2017, pp. 353–365.

[31] T. Finch, "Incremental calculation of weighted mean and variance," *University of Cambridge*, vol. 4, no. 11-5, pp. 41–42, 2009.

[32] E. Even-Dar and Y. Mansour, "Learning Rates for Q-learning," *Journal of Machine Learning Research*, vol. 5, no. Dec, pp. 1–25, 2003.

[33] K. Asadi, D. Misra, and M. L. Littman, "Lipschitz continuity in model-based reinforcement learning," *arXiv preprint arXiv:1804.07193*, 2018.

Impact of Traffic Lights on Trajectory Forecasting of Human-driven Vehicles Near Signalized Intersections

Geunseob (GS) Oh, Huei Peng

Abstract— **Forecasting trajectories of human-driven vehicles is a crucial problem in autonomous driving. Trajectory forecasting in the urban area is particularly hard due to complex interactions with cars and pedestrians, and traffic lights (TLs). Unlike the former that has been widely studied, the impact of TLs on the trajectory prediction has been rarely discussed. Our contribution is twofold. First, we identify the potential impact qualitatively and quantitatively. Second, we present a novel resolution that is mindful of the impact, inspired by the fact that human drives differently depending on signal phase and timing. Central to the proposed approach is *Human Policy Models* which model how drivers react to various states of TLs by mapping a sequence of states of vehicles and TLs to a subsequent action of the vehicle. We then combine the Human Policy Models with a known transition function (system dynamics) to conduct a sequential prediction; thus our approach is viewed as *Behavior Cloning*. One novelty of our approach is the use of vehicle-to-infrastructure communications to obtain the future states of TLs. We demonstrate the impact of TL and the proposed approach using an ablation study for longitudinal trajectory forecasting tasks on real-world driving data recorded near a signalized intersection. Finally, we propose *probabilistic* (generative) Human Policy Models which provide probabilistic contexts and capture competing policies, e.g., *pass* or *stop* in the yellow-light dilemma zone.**

I. Introduction

Autonomous driving has been more successful in highway than in urban city mainly due to the simplicity of its driving environment; absence of traffic signals, and more stable interactions with other vehicles. Realizing fully autonomous vehicles in urban driving environments is more challenging for the opposite reasons.

One of the major differences between urban city and highway driving is traffic lights (TLs). In urban areas, especially in the vicinity of TLs exemplified by signalized corridors or intersections, the motions of vehicles are mainly governed by traffic signals. People obey the traffic signals and properly respond to implicit rules imposed by traffic lights. Examples of the implicit traffic rules include stopping for a traffic light in a red phase, maintaining a proper speed in a green phase in a free-flow situation. This is why predicting how human drivers respond to traffic signals is the key to the trajectory forecasting in urban area. The decision-making, path planning, and control synthesis all benefit from more accurate trajectory forecasting, ultimately leading to successful autonomous driving.

Recent studies in trajectory forecasting utilize generative models (e.g., variational autoencoders (VAE) [1] or generative adversarial networks (GAN) [2], [3]) or convolutional & recurrent neural-nets models [4], [5], [6], [7], [8], [9]. The existing works mainly focus on accurate modeling of the interactions among vehicles [1], [4], [5], [6], [10], and/or pedestrians [2], [7], [8], [9]. Despite the important role of TLs in the vehicle motions, efforts to understand the dynamics between TLs and vehicle trajectories and to quantify the impacts of TLs on the trajectories have barely been made in the trajectory forecasting literature.

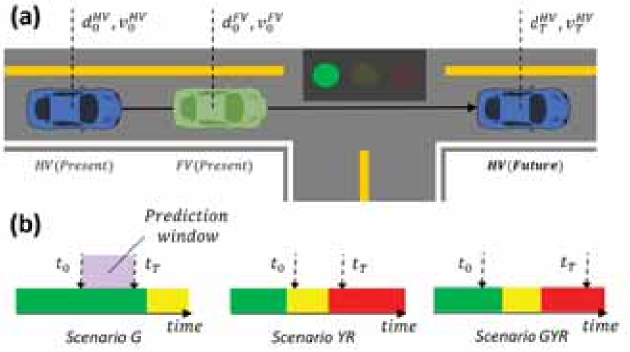


Fig. 1. (a) The trajectory forecasting problem near a traffic light for the vehicles with *through* moves is depicted. Given a sequence of past states of a host vehicle (HV) and contexts (states of TLs, its front vehicle (FV)), our goal is to forecast HV's states under various states of TLs. (b) Three example scenarios of the problem are described. The full list of the scenarios is described in Table. I. We define 'scenario G' as a forecasting problem when the prediction window starts on a green light and ends on the same green light. 'Scenario GYR' represents a forecasting problem where the window spans over a set of green, yellow, and red lights.

The less recognized impact of TLs in the vehicle trajectory forecasting is elaborated in Fig. 2. Specifically, Fig. 2 depicts an example which shows how the states of TLs affect vehicle trajectories and how uncertainties in the states of TLs can cause high prediction errors. Even for models that account for the uncertainties in the future phase (it can either be red or green) and output probabilistic predictions for the two possibilities (either the phase remains red, or the phase shifts to green), the uncertainty question still is not resolved: precisely when will the phase change?

On the other hand, there have been efforts to model the dynamic impact of TLs in the transportation research community. However, there is no comprehensive model that describes behavior of human drivers near traffic signals. A few papers have studied specific instances of the dynamics but limited to a few simple scenarios; [11], [12], [13] developed models for vehicles approaching a signalized intersection (SI) and making complete stops in red light. [13], [14], [15], [16], [17] proposed models for vehicles departing

Geunseob (GS) Oh and Huei Peng are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA, Email : {gsoh, hpeng}@umich.edu

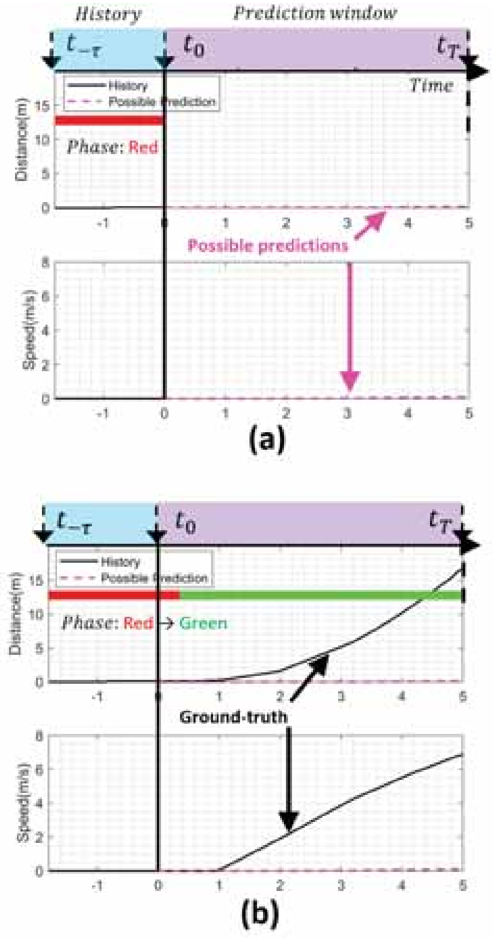


Fig. 2. A motivational example from our real-world dataset is depicted to demonstrate how uncertainty in the states of TLs makes the trajectory forecasting difficult. The sample is a 7s long RG scenario, where the first 2s is the observed history and the last 5s is the future. The goal is to predict *longitudinal* position and speed of a vehicle for $t = 0s : 5s$. We define $0m$ as the position of the vehicle at $t = 0s$. The distance to TL is then $13m$. (a) Given the history of the vehicle's position, speed, and the red phase observed at $t = 0s$, a reasonable prediction under uncertain traffic phase at $t > 0s$ that an existing method would make is to predict the vehicle to stay put (pink dotted line). (b) However, in reality, the phase shifted to green at $t = 0.35s$ and the vehicle sped away as depicted as black line, resulting in large prediction errors in both position and speed.

from a SI from zero-speed in green phase. These models are either limited to specific instances of the problem, or are not forecasting algorithms since they require parameters like total deceleration time, final speed, which can only be measured after a trip is complete. [18], [19] presented prediction algorithms for the vehicles in highway and in car-following scenarios [20] based on car models proposed in [21], [22]. These models, however, do not describe how human drivers react to the traffic signals.

In order to leverage rich information that comes from the dynamics between TLs and human drivers and to apply it to the trajectory forecasting problem, we simply utilize vehicle-to-infrastructure (V2I) communications. By leveraging V2I communication, one can access **the future profile** of the TLs ahead of time, which resolves the aforementioned uncertainty problem inherent to any prediction task near traffic lights. Based on this simple but novel idea, we investigate how the access to the future profile of the TLs can improve the accuracy of the trajectory forecasts.

In this paper, we tackle a trajectory forecasting problem described in Fig. 1. In order to solely focus on the impact of TLs, the forecasting task is simplified to a longitudinal trajectory forecasting with a front vehicle (FV) near a TL. This setting assumes that the impact of rear & side vehicles is minimal and no vehicle cuts into the lane that HV is located.

Our solution approach consists of two models which map a sequence of states of a HV, its FV and the corresponding states of a TL to the subsequent action (longitudinal acceleration) that the HV takes. We name the models *Human Policy Models* to highlight that the models return an action given a state. Since the models are learned to mimic drivers, the proposed approach can be viewed as *imitation learning*, or precisely *behavior cloning*, as we work with real data (i.e., no access to environment) and assume that inputs are i.i.d.

The first model is *deterministic* human policy model which returns the most-probable action and is designed using RNN-based network. The second model is *probabilistic* (generative) human policy model which outputs distribution parameters and is able to generate trajectories by sampling from the distribution and is designed using a RNN-based mixture density network (MDN) [23]. Then, we utilize these models to forecast longitudinal trajectories of the HV sequentially over a time span using the system dynamics (trasition function). For the training, validation, and testing, we used 502,253 sequences excerpted from naturalistic (non-obstructive, uncontrolled) trips from 50 distinct cars over 2 years at a signalized intersection in Ann Arbor, MI.

The remainder of the paper is organized as follows: Section II elaborates on the proposed human policy models. Section III describes the framework is used to obtain the forecast trajectories. Section IV presents results using an ablation study. Finally, Section V offers concluding remarks.

II. Human Policy Model

A. Problem Description

The problem of longitudinal trajectory forecasting under various phases and timings of TLs (Fig. 1) is challenging due to the stochastic reactions of human drivers. For example, a driver may prefer late hard-braking approaching to a red light, while others may prefer soft-braking. Also, the reactions of drivers at steady phases (i.e., green (G), yellow (Y), red (R)) are different from those at phase transitions (GY, YR, RG). Another example is the decision making in yellow light dilemma zone [24], where a driver arrives at a TL at a high speed. There usually exists two competing decisions; a driver could either engage in a hard-braking to stop before the TL or pass through the TL.

In this sense, we break down the problem to six distinct scenarios depicted in Fig. 1(b) and Table. I. The idea behind this categorization is our belief that humans react differently to various phases of TLs, resulting in unique trajectories. In this regard, we argue that a *comprehensive* model should be validated against all the 6 scenarios.

B. Related Works

To the best of our knowledge, none of the existing works in machine learning community addressed this particular forecasting problem (see Section I). In traditional transportation community, a few papers have discussed acceleration models or speed profiles near traffic signals. [13], [15], [16], [17] proposed polynomial speed & acceleration models for vehicles departing from a SI from zero-speed in G phase. [11], [12], [13] developed deceleration models for vehicles make complete stops at a TL in R phase. However, these models studied very specific instances of the problem, thus do not qualify as comprehensive model. We classified the available studies in Table. I.

TABLE I

Six distinct scenarios of the prediction problem

Scenario		Available Studies in Transportation Community
G	D0 (departure from zero-speed)	ATL Newzealand(1990), Bham(2002), Day(2013), Modified IDM(2018)
	General	None
Y		None
R	A0 (arrival to zero-speed)	Bennett(1995), Wang(2005), Modified IDM(2018)
	General	None
GY		None
YR		None
RG		None

We believe that a general model which captures the behavior of human drivers in all scenarios described in Table. I is crucial to accurate forecasting of human vehicles near TLs. To the best of our knowledge, such model does not exist.

C. Proposed Model

The key behind modeling a driver's reaction to TL is feature selection and model design so that the reaction to TL is well captured in corresponding state space. We first introduce the model, define the state, and move on to the explanation of the intuition behind the selections.

Human Policy Models are functions that map a sequence of past states of a HV ($X^{HV}_{t-\tau:t} := [d_{t-\tau:t}, v_{t-\tau:t}]$) and a context vector ($C_t := [X^{FV}_t, X^{TL}_t, TOD_t]$) to a subsequent longitudinal acceleration of HV (a^{HV}_t). d_t and v_t each indicates the distance to the traffic light and speed at time t. C_t includes state of the FV at time t ($X^{FV}_t := [FV_t, r_t, \dot{r}_t]$) where FV_t is a binary flag which represents the presence of a FV, and $r_t, \dot{r}_t$ indicate positions and speed relative to the HV. The state of the corresponding TL at time t ($X^{TL}_t := [P_t, T_t]$) is defined as phase (G,Y,R) as P_t and timing (i.e., time elapsed in the current phase) of TLs as T_t. TOD_t is the time of day at time t. The intuitions behind the selection of the input features ($X_t := [X^{HV}_{t-\tau:t}, X^{FV}_t, X^{TL}_t, TOD_t]$) are as follows.

Distance to traffic light (d_t), **Speed** (v_t) each represents the longitudinal distance of a HV to the TL that the HV is approaching or departing from and the longitudinal speed of the HV. They are essential in forecasting vehicle trajectories near TLs. For example, a HV approaching a TL in red phase travels slowly when it is close to the TL, whereas it can travel fast when it is far away from the TL. $d_x > 0$ means that the HV is approaching the TL, and $d_x < 0$ indicates that it's departing from the TL. We assume $v_x >= 0$.

Range and range-rate ($r_t, \dot{r}_t$) represent the longitudinal position and speed of the FV relative to those of the HV. We assume $r_x > 0$, meaning that the FV is always ahead of the HV. Note, rear & side vehicles are not considered due to the dataset availability. However, one can trivially extend our model to include them.

Phase and timing of traffic light (P_t, T_t) represents the phase of a TL (G,Y,R) that a HV is subject to and the time elapsed since the last phase change ($T_t >= 0$). T_t accounts for transient behaviors of human drivers at phase shifts. For example, a vehicle approaching a TL in a red phase with a small T_t, meaning that the phase just shifted to red, may not be traveling slowly whereas a vehicle approaching a TL with a large T_t is likely to travel slowly or fully stopped.

Time of day (TOD) represents the time of day as elapsed hours since midnight ($0 \leq TOD < 24$). TOD=12 indicates noon. Macroscopic traffic characteristics including traffic speed differ considerably depending on TOD as evidenced in studies including [25]. The selection of TOD is an attempt to incorporate the macroscopic trend of the traffic.

Due to stochastic and complex nature of human decision making in driving, a simple analytical model such as a polynomial or a physics-based model may not represent the nominal or probabilistic behaviors of human-drivers near traffic signals well. This is why we opt for data-driven approach.

D. Dataset

The real-world driving data utilized in this work are from Safety Pilot Model Deployment (SPMD), a large-scale connected vehicle study conducted in Ann Arbor, MI [26]. The vehicles were equipped with data loggers which collected 10Hz GPS signals including coordinates, speeds, and heading angles as well as 10Hz front vehicles data such as relative positions and speeds. While SPMD database does not include any vision data (e.g., lidar, camera, or radar) nor the access to all vehicles in the scene, it provides detailed TL profiles that were obtained using V2I communication devices installed at signalized intersections (SIs) and information about vehicles in front of HV. To the best of our knowledge, no dataset is publicly available that provides the detailed TL data. Hence, we leverage SPMD for its unique access to the TL profiles.

In this work, we extracted 502,253 observations (samples) from 50 distinct SPMD vehicles collected over a span of 27 months (Mar 2015 - July 2017) near the Plymouth Rd & Huron Pkwy intersection. Each observation was synchronized with the traffic signal states of the SI. In order to reduce the noise in $X^{HV}_t, X^{FV}_t, a^{HV}_t$, a least-square polynomial smoothing filter was used [27].

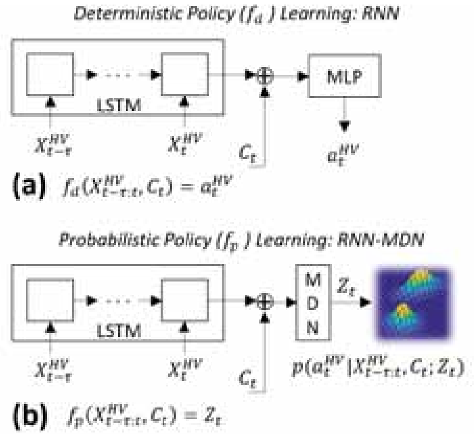

Deterministic Policy (f_d) Learning: RNN
LSTM
MLP
$X^{HV}_{t-\tau}$
X^{HV}_{t}
C_t
a^{HV}_t
(a) $f_d(X^{HV}_{t-\tau:t}, C_t) = a^{HV}_t$
Probabilistic Policy (f_p) Learning: RNN-MDN
LSTM
M
D
N
Z_t
$X^{HV}_{t-\tau}$
X^{HV}_{t}
C_t
$p(a^{HV}_t | X^{HV}_{t-\tau:t}, C_t; Z_t)$
(b) $f_p(X^{HV}_{t-\tau:t}, C_t) = Z_t$

Fig. 3. The proposed policy models are described. (a) a deterministic policy ($f_d : [X^{HV}_{t-\tau:t}, C_t] \rightarrow a^{HV}_t$), (b) a probabilistic policy ($f_p : [X^{HV}_{t-\tau:t}, C_t] \rightarrow Z_t$).

E. Implementation Details

Both the deterministic (f_d) and probabilistic (f_p) human policy models were implemented in Tensorflow-Keras. Each model consists of a double-stacked LSTM which takes $X^{HV}_{t-\tau:t}$ followed by a concatenation with the context vector $C_t = [X^{FV}_t, X^{TL}_t, TOD_t]$. The concatenated tensor is then fed into a multi-layer perceptron (MLP) for the deterministic model or a MDN for the probabilistic model. The MLP layer outputs a^{HV}_t whereas the MDN layer outputs the distribution parameters Z_t. As we model MDN using gaussian mixtures, the MDN layer outputs three sets of the parameters: mixture weights π_k, means μ_k, variances σ^2_k for N components. We used $N = 2$ for the yellow light dilemma scenario. Models were trained using ADAM optimizer.

f_d is learned by minimizing a loss function L_d which is a summation of mean squared error as described below.

$$L_d := \sum_{t=1}^{T} (a^{HV}_t - f_d(X^{HV}_{t-\tau:t}, C_t))^2 \tag{1}$$

f_p is obtained by minimizing a loss function L_p which is a sum of a negative log-likelihood.

$$L_p := \sum_{t=1}^{T} -log(p(a^{HV}_t | X^{HV}_{t-\tau:t}, C_t; Z_t)) \tag{2}$$

III. Trajectory Forecast Framework

Fig. 4 illustrates the proposed framework that consists of two parts. The first part is an off-line supervised learning of the policies. The second part is where we use the learned policies in sequential prediction setting to obtain $X^{HV}_{0:T}$ (i.e., alternation of one-step predictions of the learned policies and state transitions). The system dynamics (i.e., transition function for X^{HV}) is given as longitudinal vehicle kinematics with a zero-order hold, as described in Eq. 3 and 4.

$$v_{n+1} := v_n + a_n \Delta t_n, d_{n+1} := d_n + 0.5(v_{n+1} + v_n)\Delta t_n \tag{3}$$

$$X^{HV}_{n+1} = A_n X^{HV}_n + B_n a^{HV}_n \tag{4}$$

where $A_n := \begin{bmatrix} 1 & \Delta t_n \\ 0 & 1 \end{bmatrix}, B_n := \begin{bmatrix} 0.5\Delta t_n \\ \Delta t_n \end{bmatrix}, X^{HV}_n = \begin{bmatrix} d_n \\ v_n \end{bmatrix}$.

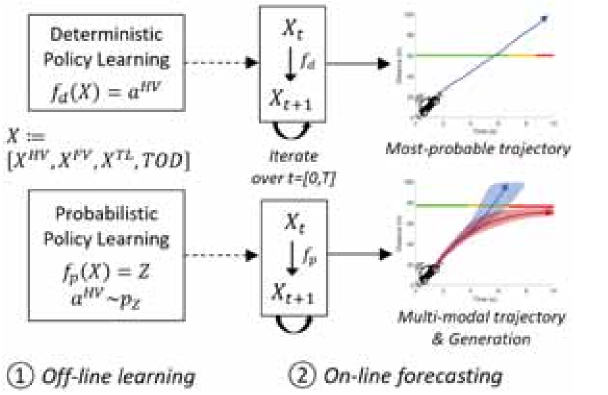


Fig. 4. The framework is divided into two steps. The first step is the off-line training of the proposed models. The second step involves propagations of the policies and state transitions to obtain vehicle trajectories over T.

Defining $n = 0$ and $n = N$ as the index for $t = 0$ and $t = T$, the trajectory forecast over the prediction horizon $t = [0, T]$ are obtained by propagating Eq. 4 from $n = 0$ to $n = N - 1$:

$$\begin{aligned} X^{HV}_N &= \prod_{k=0}^{N-1} A_k X^{HV}_0 + \prod_{k=1}^{N-1} A_k B_0 a^{HV}_0 + \prod_{k=2}^{N-1} A_k B_1 a^{HV}_1 \\ &\quad + \ldots + A_{N-1} B_{N-2} a^{HV}_{N-2} + B_{N-1} a^{HV}_{N-1} \\ &= \prod_{k=0}^{N-1} A_k X^{HV}_0 + \prod_{k=1}^{N-1} A_k B_0 f(X^{HV}_{-n_\tau+1:0}, C_0) \\ &\quad + \ldots + B_{N-1} f(X^{HV}_{N-n_\tau:N-1}, C_{N-1}) \\ &:= F(X^{HV}_{1:N-1}, X^{HV}_{-n_\tau+1:0}, C_{0:N-1}, \Delta t_{0:N-1}) \end{aligned} \tag{5}$$

where f can either be f_d or $S(f_p)$. S is a function which returns a sample a^{HV}_t from the pdf. In case of 1D Gaussian, $Z_t := [\mu_t, \sigma^2_t]$ and $a^{HV}_t \sim N(\mu_t, \sigma^2_t)$. τ, n_τ each indicates input sequence length in time and in the number of steps.

As described in Eq. 5, X^{HV}_N is a function (F) of $[X^{HV}_{1:N-1}, X^{HV}_{-n_\tau+1:0}, C_{0:N-1}, \Delta t_{0:N-1}]$. The second term $X^{HV}_{-n_\tau+1:0}$ is given and the last term $\Delta t_{0:N-1}$ can simply be predetermined based on a required time resolution. Obtaining $C_{0:N-1}$ at the prediction time ($t = 0$) is the main challenge, due to uncertainties in $X^{FV}_{1:N-1}, X^{TL}_{1:N-1}$. A simple way to get away with the uncertainties is to design a model to predict trajectories ($X^{HV}_{0:T}$) conditioned only on the observed states $[X^{FV}_{-\tau:0}, X^{TL}_{-\tau:0}]$. An example is a model with many-to-many RNN that takes a sequence of past states and returns a sequence of future states; which is a forecasting model that does not utilize the future states of TLs. Fig. 2 showed how such model can fail.

This is where our novel idea comes into play. We remove the uncertainties by utilizing the future phases and timings of TLs obtained via V2I communications. With the access, $X^{TL}_{1:N-1}$ can be attained at the prediction time. The remainder is then $X^{FV}_{1:N-1}$, which is obtained using a variant of f_d. Specifically, we train another human policy model f^{NoFV}_d : $[X^{HV}_{N-n_\tau:N}, C'_N] \rightarrow a^{HV}_N$ with $C'_N := [X^{TL}_N, TOD_N]$ excluding X^{FV}_N from C (i.e., f^{NoFV}_d does not consider FV). After the off-line learning of f^{NoFV}_d, we apply the aforementioned iterative process on FV to obtain $X^{FV}_{1:N-1}$ via Eq. 5 with f^{NoFV}_d.

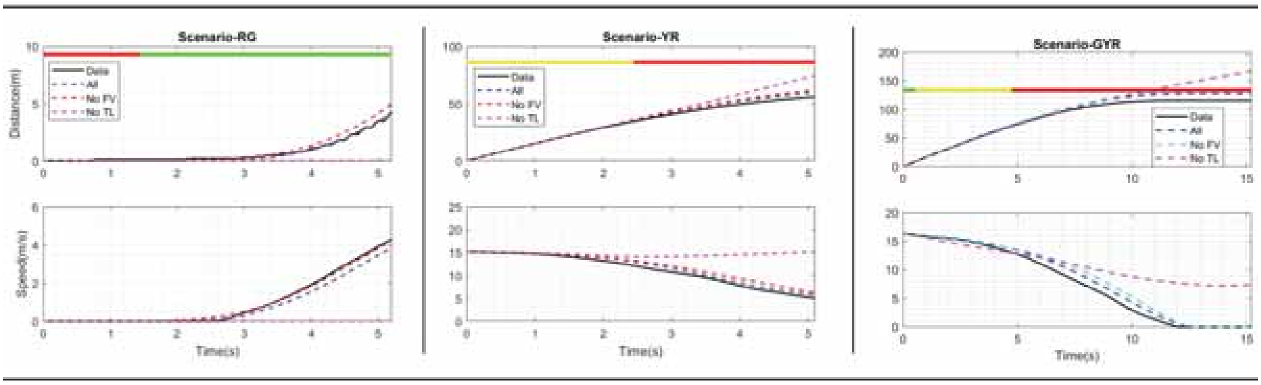


Fig. 5. For qualitative evaluations, ground-truth and predicted trajectories are depicted for two 5s sample scenarios RG (left), YR (middle), and one 15s sample scenario GYR (right). Four trajectories of distance to the TL (1st row), speed (2nd row) are shown in each plot and represent ground-truth (black), predictions $X^{HV}_{0:5s}$ obtained using f_d(Blue), f_d^{NoFV}(Red), f_d^{NoTL}(Pink). The 2s history $X^{HV}_{-2:0s}$ (i.e., part of inputs to the models) are omitted for the simplicity.

Once $X^{FV}_{1:N-1}, X^{TL}_{1:N-1}$ are secured, the resulting trajectory forecast $X^{HV}_{1:N}$ is obtained via Eq. 5. Since $X^{HV}_{1:N}$ can simply be forecast using f_d^{NoFV}, we conduct an ablation study (elaborated in Section IV) on f_d, f_d^{NoFV} and the other two models (f_d^{NoTL}, f_d^{NoFVTL}) which each represents unique model where X^{TL} and $[X^{FV}, X^{TL}]$ are excluded from C.

For the probabilistic human policy model, the probability of the resulting trajectory forecast $p(X^{HV}_{1:N})$ can be estimated using the chain rule of probability, which factorizes the joint distribution over N separate conditional probabilities:

$$p(X^{HV}_{1:N} | X^{HV}_{-n_\tau+1:0}, C_{0:N-1}, \Delta t_{0:N-1}) = \prod_{k=1}^{N} p(X^{HV}_k | X^{HV}_{1:k-1}, X^{HV}_{-n_\tau+1:0}, C_{0:k-1}, \Delta t_{0:k-1}) \tag{6}$$

As opposed to the deterministic forecasting where the most-probable trajectory is obtained, a resulting trajectory is a sample from a probability distribution. While we can estimate the joint probability density of a trajectory forecast via Eq. 6, the marginal probability of X^{HV}_t needs to be numerically estimated via sampling since the distribution parameter Z_t is obtained via an arbitrarily complex neural network and depends on previous predictions $X^{HV}_{-n_\tau+1:k-1}$. Thus, we utilize Monte Carlo Simulation to obtain the samples (roll-out trajectories) and kernel density estimation to approximate the marginal probability density of the samples.

IV. Results

In this section, we discuss evaluation results conducted on the dataset (see Section II-D). In Section IV-A, we present the impact of X^{TL} qualitatively by comparing the trajectory forecasts made using the four deterministic policies and that the utilization of phases and timings of TLs helps forecasting trajectories more accurately. In Section IV-B, we discuss a set of metrics to evaluate performance of the forecasts. The ablation study in Section IV-C is designed to evaluate the 4 variants of our deterministic models ($f_d, f_d^{NoFV}, f_d^{NoTL}, f_d^{NoFVTL}$) by quantifying their performance using 3,111 test snippets. In Section IV-D, we demonstrate how the proposed probabilistic models can tackle scenarios with competing policies.

A. Impact of X^{TL} on trajectory forecasts

As explained in Section III, we first train 4 distinct deterministic policy models $f_d, f_d^{NoFV}, f_d^{NoTL}$, and f_d^{NoFVTL}. The superscript specifies the context input C that the model takes (i.e., $C^{NoFVTL} := [TOD], C^{NoTL} := [X^{FV}, TOD], C^{NoFV} := [X^{TL}, TOD]$, where C^{mode} is the context input for f_d^{mode}). Each model then produces unique trajectory forecasts per scenario. Three scenarios (RG, YR, GYR) are sampled and the four models are used to forecast trajectories. In each sample, four trajectories are depicted to represent ground-truth and 3 predicted trajectories each from f_d, f_d^{NoFV}, f_d^{NoTL}.

The sample on the left of Fig. 5 is similar to the motivational example in Fig. 2. At $t = 0$, the driver was at a stop at a red phase. A reasonable prediction that a model without $X^{TL}_{0:5s}$ would make is to forecast the vehicle to stay put. As expected, f_d^{NoTL} (pink) failed to predict X^{HV} accurately, whereas the forecasts from the other two models f_d, f_d^{NoFV} which utilize $X^{TL}_{0:5s}$ are close to the ground-truth.

The middle plot of Fig. 5 is a scenario YR where the driver was cruising as approaching the TL ($v^{HV}_{-2:0s} = 15$). Given $P_0 = Y$ and that the vehicle was cruising, f_d^{NoTL} forecast the vehicle to maintain the speed, causing the prediction errors to grow over time. The other two models f_d, f_d^{NoFV} which use $X^{TL}_{0:5s}$ took account for the phase shift in the future and accurately forecast how the driver would react to the shift.

On the other hand, the right plot serves as an exemplar long-term scenario. A 15s scenario that spans a full cycle of phases (GYR) is illustrated where HV was initially decelerating while approaching the TL. All models captured temporal trends in the speed and predicted that HV would continue to decelerate. f_d^{NoTL} predicted that the vehicle would cross the intersection, given $P_0 = G$ whereas f_d, f_d^{NoFV} predicted that the vehicle would make a stop before the TL considering the future states of TL. Indeed, the ground-truth is that the vehicle made a stop before the TL to react to the phase shift.

As demonstrated, the impact of X^{TL} is significant; uncertainties in X^{TL} can cause high prediction errors, especially for long-term predictions. The results suggest that forecasting methods may perform poor without knowledge of future X^{TL}, highlighting why the problem is critical. While information

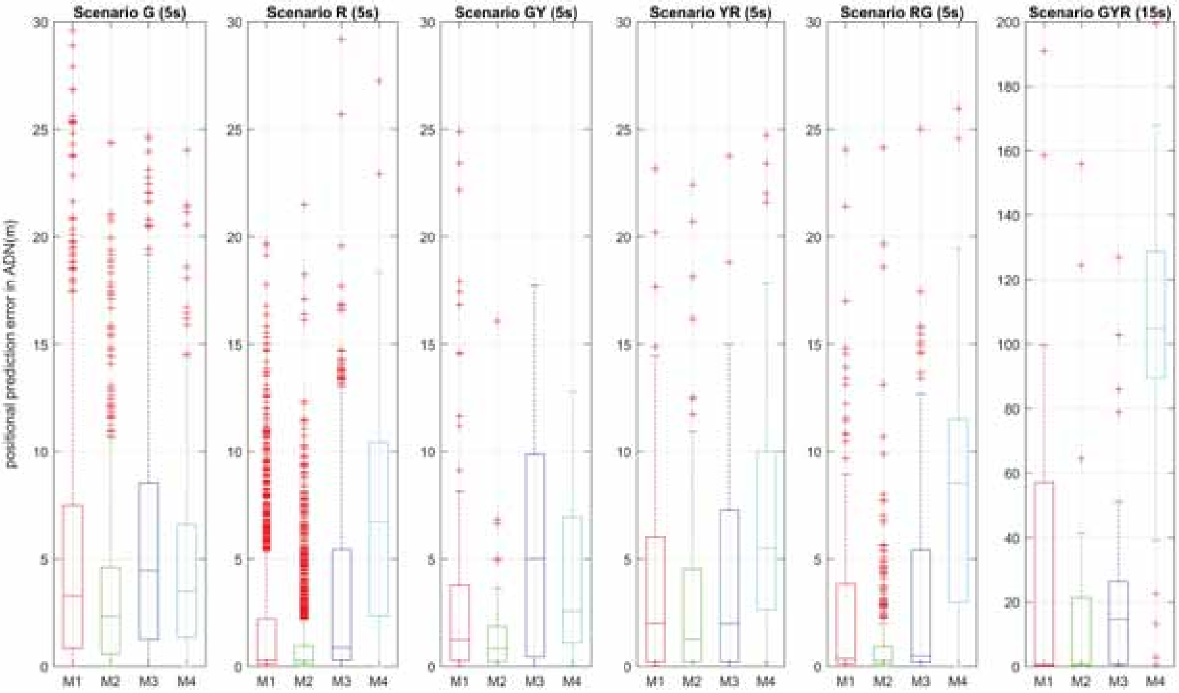


Fig. 6. A quantitative evaluation is conducted on the test set and depicted as boxplots for the six scenarios using the evaluation metric *ADN*. M1, M2, M3, M4 each represents forecasting models with $f_d, f_d^{NoFV}, f_d^{NoTL}, f_d^{NoFVTL}$. The prediction window is $T = 15s$ for scenario GYR, and $T = 5s$ for others.

of TLs can be transmitted through FV to some extent, such transmission is ineffective when (1) FV is absent, (2) FV is present, but, far from HV, or (3) FV is present and close to HV, however, the phase transition is imminent. The proposed idea is a solution to the problem: utilization of the future X^{TL} greatly improves the quality of forecasts near traffic lights.

B. Evaluation metrics

We use the following metrics for the quantitative evaluation: mean absolute error (MAE), time weighted absolute error (TWAE), absolute deviation at the end of the prediction window (ADN) defined in Eq. 7, 8, and 9, where $\hat{X}_k^{HV}, X_k^{HV}$ represents the kth-step forecast X and ground truth X.

$$MAE := \frac{\sum_{k=1}^{N} |\hat{X}_k^{HV} - X_k^{HV}|}{N} \tag{7}$$

$$TWAE := \frac{\sum_{k=1}^{N} (t_k |\hat{X}_k^{HV} - X_k^{HV}|)}{\sum_{k=1}^{N} t_k} \tag{8}$$

$$ADN := |\hat{X}_N^{HV} - X_N^{HV}| \tag{9}$$

We used $\forall k : \Delta t_k = 0.2s$, $\tau = 2s$ (history). For the scenario with a prediction window $t_N = 5s$, the last index N is 25.

C. Ablation Study

The goal of the ablation study is to quantify the impacts of X^{TL} on a large testset and to evaluate performance of the four policies. The result (on ADN) is presented in Fig. 6. The sample sizes are 688 (G), 1909 (R), 68 (GY), 81 (YR), 362 (RG), 32 (GYR), totalling 3,111 sample snippets. The detailed results for the ablation study on all three metrics (MAE, TWAE, ADN) are presented in Table II, III, each for position and velocity errors. Note, scenario Y is not depicted due to its short (and inconsistent) prediction horizon; we observed that the phase Y usually lasts anywhere between 2.5s to 4s. The first 2s are used as inputs, which means the prediction horizon for the scenario Y is only 0.5s to 2s.

As depicted in Table II, III, the magnitude of error is as follows: MAE<TWAE<ADN, making ADN the largest error. As shown in Fig. 6, across all scenarios, the two models f_d, f_d^{NoFV} which utilizes X^{TL} outperform the other two models f_d^{NoTL}, f_d^{NoFVTL} which don't take advantage of the future phases and timings information. Interestingly, the winner is not f_d, but it is f_d^{NoFV}, which performs the best on all characteristics of boxplot including the 1st, 3rd quartiles, the median, and the upper limit of the extreme points. Our interpretation is that the exclusion of X^{FV} from C increases the prediction accuracy, due to the uncertainty in $\hat{X}_{t>0}^{FV}$ as $\hat{X}_{t>0}^{FV}$ are predicted via another human policy model.

The numbers presented in Table II, III agree with the results from Fig. 6 across all scenarios. f_d^{NoFV} is the winner for almost all metrics, or at least on par with f_d. In summary, the knowledge of future states of traffic lights significantly increase the accuracy of trajectory forecasts, as evidenced in the ablation studies: trajectory forecasts with the winner model have roughly 1.5-30 times smaller (position) MAE, TWAE, ADN for $T = 5s$ scenarios (G, R, GY, YR, RG), and roughly 9-150 times smaller MAE, TWAE, ADN for $T = 15s$ scenario (GYR), compared to trajectory forecasts via f_d^{NoTL}, f_d^{NoFVTL}. This discrepancy in the accuracy becomes bigger as the prediction horizon grows, especially for the long-term forecasts such as scenario GYR.

TABLE II
Ablation study on **position** errors with average MAE, TWAE, ADN. Prediction horizon is 15s for GYR and 5s for others. The lower a metric is the better. The numbers from the best performing model are marked in **bold**.

Scenario	MAE(m)				TWAE(m)				ADN(m)			
	All	NoFV	NoTL	NoFVTL	All	NoFV	NoTL	NoFVTL	All	NoFV	NoTL	NoFVTL
G	1.18	**0.78**	1.35	1.19	1.86	**1.21**	2.16	1.78	3.28	**2.34**	4.47	3.51
R	0.16	**0.15**	0.38	1.79	0.21	**0.20**	0.56	2.98	0.31	**0.28**	0.88	6.74
GY	0.53	**0.41**	1.26	0.82	0.77	**0.54**	2.06	1.31	1.24	**0.84**	5.01	2.59
YR	1.18	**0.69**	0.86	1.01	1.60	**0.93**	1.32	1.42	2.00	**1.27**	1.98	2.29
RG	0.17	**0.14**	0.22	2.28	0.24	**0.19**	0.31	3.80	0.36	**0.28**	0.50	8.53
GYR	**0.445**	**0.445**	4.11	32.40	**0.567**	0.568	6.82	51.16	**0.694**	**0.694**	14.65	104.81

TABLE III
Ablation study on **velocity** errors with average MAE, TWAE, ADN. Prediction horizon is 15s for GYR and 5s for others. The lower the better.

Scenario	MAE(m/s)				TWAE(m/s)				ADN(m/s)			
	All	NoFV	NoTL	NoFVTL	All	NoFV	NoTL	NoFVTL	All	NoFV	NoTL	NoFVTL
G	0.70	**0.46**	0.97	0.75	1.02	**0.65**	1.41	1.07	1.48	**0.94**	2.28	1.51
R	**0.005**	**0.005**	0.17	1.44	0.007	**0.006**	0.22	2.29	0.010	**0.009**	0.29	4.10
GY	0.39	**0.15**	1.11	0.53	0.54	**0.20**	1.78	0.79	0.87	**0.27**	3.08	1.21
YR	**0.367**	0.374	0.53	1.01	**0.47**	0.52	0.71	1.42	**0.59**	0.69	0.89	2.29
RG	0.007	**0.005**	0.10	1.68	0.010	**0.007**	0.14	2.63	0.02	**0.01**	0.22	4.52
GYR	0.015	**0.014**	1.26	6.78	**0.018**	0.020	1.86	9.80	0.030	**0.029**	2.60	14.13

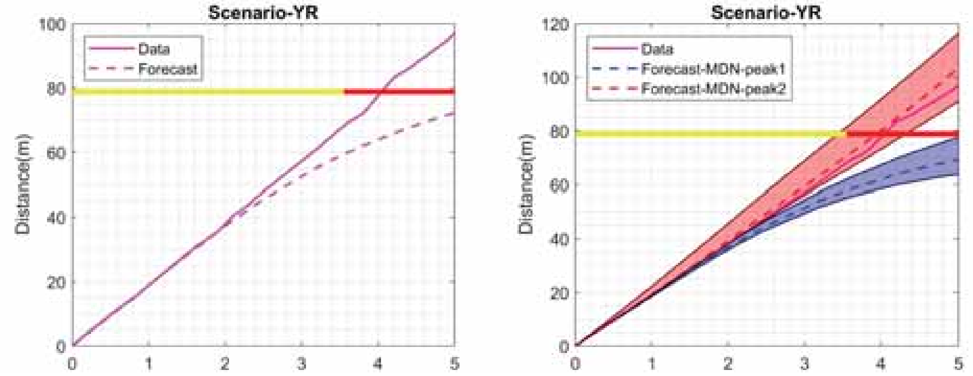


Fig. 7. A sample trip for the yellow light dilemma scenario. The left plot highlights the limitation of the most-probable trajectory. The right plot shows that the proposed probabilistic models are able to forecast the two competing policies. The trajectories $\forall t : p(d_t) >= 0.01$ are illustrated.

D. Probabilistic Prediction

The outliers observed (indicated as '+') in Fig. 6 occur mostly from edge cases and competing policies. Examples of the edge cases include a driver approaching a TL in $P_{\forall t} = R$ with high speed and executing a sudden break right before the TL rather than gradually slowing down as it approaches the intersection. Another example is that a driver in the middle of the road in $P_{\forall t} = G$ moving much slower than the average speed of the traffic for unknown reasons. The outliers occur from competing policies are exemplified by the yellow light dilemma scenario where a driver can either cross the intersection or stop before the intersection.

Fig. 7(a) describes a sample trip observed in our dataset that represents the yellow light dilemma scenario. As shown in the figure, the most-probable trajectory forecast obtained via f_d^{NoFV} predicts that the vehicle would make a stop before the intersection, however, the driver crossed the intersection even after the phase shifted to red.

This is where the proposed probabilistic models come in handy. As the probabilistic model is capable of reproducing multi-modal distributions, it captures the other competing policy (*cross*). In addition, it is able to reason uncertainties of the forecasts as the probabilities of the forecasts can be estimated via Monte-Carlo simulation. Another advantage is that we can generate (sample) possible trajectories as the proposed probabilistic policy outputs both likelihood $p(X|Z)$ and prior $p(Z)$; i.e., the model is generative.

We argue that the deterministic models are still valuable:

the solutions are simple, cost-efficient, and easy to interpret. They can serve as nominal trajectories of human drivers in situations that can be approximated uni-modal. The nominal trajectories can be used in a trajectory planning algorithm which works with deterministic actors or disturbances.

For the scenarios with 5s prediction horizon, the time to compute a most-probable trajectory forecast is less than 10milliseconds on a single-core personal laptop with i7-6500U 2.50GHz CPU, and 8GB RAM without utilizing a parallelization. However, it takes several seconds (5-10s for 1,000 rollout trajectories) to construct the pdf for the probabilistic forecasts on the same machine due. One can significantly reduce the time via parallel computing (GPU).

V. Conclusion

Our work is the first attempt in the community to comprehensively understand and identify the impact of traffic signals on trajectory forecasting near TLs. In this regard, we first introduced the motivational example in Fig. 2 and defined 6 distinct scenarios of the problem. We proposed a novel idea to solve the scenarios that leverages the **future states** of traffic lights obtained via V2I communications. Specifically, we proposed deterministic and probabilistic human policy models to simulate state-dependent driver actions near TLs. In the ablation study, we show that the utilization of future phases and timings of TLs significantly improves the quality of trajectory forecasts for all scenarios described in Table. I.

As no dataset is publicly available that has the detailed TL data, interactions with surrounding vehicles, and the vision data, our experiments were based on a non-vision dataset with or without a front vehicle in the scene. Hence, a direct comparison against state-of-the-art forecasting models [1], [2], [3], [7], [8] is not made since the models were built on the vision data and/or with the access to all vehicles adjacent to the host vehicle (e.g., side or rear cars). Regardless, the proposed idea and the proposed framework can be leveraged by any work that concerns vehicle trajectory forecasting near TLs given the access to V2I communications. As the results suggest that the utilization of the future TL states could lead to the significant improvements in the prediction accuracy, we believe that it is worth building a large-scale dataset with both the TL and vision data to quantify the influence to the fullest extent.

ACKNOWLEDGMENT

The authors would like to thank the University of Michigan Transportation Research Institute for access to the data.

References

[1] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 336–345.

[2] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[3] J. Li, H. Ma, and M. Tomizuka, "Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning," *arXiv preprint arXiv:1904.02390*, 2019.

[4] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.

[5] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3960–3966.

[6] R. Chandra, T. Guan, S. Panuganti, T. Mittal, U. Bhattacharya, A. Bera, and D. Manocha, "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms," *arXiv preprint arXiv:1912.01118*, 2019.

[7] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[8] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8483–8492.

[9] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4194–4202.

[10] R. Chandra, U. Bhattacharya, C. Roncal, A. Bera, and D. Manocha, "Robusttp: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs," in *ACM Computer Science in Cars Symposium*, 2019, pp. 1–9.

[11] C. R. Bennett and R. Dunn, "Driver deceleration behavior on a freeway in new zealand," *Transportation Research Record*, no. 1510, 1995.

[12] J. Wang, K. K. Dixon, H. Li, and J. Ogle, "Normal deceleration behavior of passenger vehicles at stop sign–controlled intersections evaluated with in-vehicle global positioning system data," *Transportation research record*, vol. 1937, no. 1, pp. 120–127, 2005.

[13] C. Sun, X. Shen, and S. Moura, "Robust optimal eco-driving control with uncertain traffic signal timing," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 5548–5553.

[14] R. Akçelik and D. Biggs, "Acceleration profile models for vehicles in road traffic," *Transportation Science*, vol. 21, no. 1, pp. 36–54, 1987.

[15] G. Bham and R. Benekohal, "Development, evaluation, and comparison of acceleration models," in *81st Annual Meeting of the Transportation Research Board, Washington, DC*, vol. 6, 2002.

[16] ATS, "Acceleration/deceleration profiles at urban intersections," *Transit New Zealand Australasian Traffic Surveys*, 1990.

[17] P. P. Dey, S. Nandal, and R. Kalyan, "Queue discharge characteristics at signalised intersections under mixed traffic conditions," *European Transport*, vol. 55, no. 7, pp. 1–12, 2013.

[18] J. Park, D. Li, Y. L. Murphey, J. Kristinsson, R. McGee, M. Kuang, and T. Phillips, "Real time vehicle speed prediction using a neural network traffic model," in *The 2011 International Joint Conference on Neural Networks*. IEEE, 2011, pp. 2991–2996.

[19] B. Jiang and Y. Fei, "Vehicle speed prediction by two-level data driven models in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1793–1801, 2016.

[20] K. Fadhloun, H. Rakha, and P. Eng, "A vehicle dynamics model for estimating typical vehicle accelerations 2," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 35, no. 36, p. 37, 2015.

[21] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[22] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[23] C. M. Bishop, "Mixture density networks," Citeseer, Tech. Rep., 1994.

[24] T. J. Gates, D. A. Noyce, L. Laracuente, and E. V. Nordheim, "Analysis of driver behavior in dilemma zones at signalized intersections," *Transportation Research Record*, vol. 2030, no. 1, pp. 29–39, 2007.

[25] G. Oh, D. J. Leblanc, and H. Peng, "Vehicle energy dataset (ved), a large-scale dataset for vehicle energy consumption research," *arXiv preprint arXiv:1905.02081*, 2019.

[26] D. Bezzina and J. R. Sayer, "Safety Pilot: Model Deployment Test Conductor Team Report," Tech. Rep. June, 2015. [Online]. Available: http://safetypilot.umtri.umich.edu/

[27] R. W. Schafer *et al.*, "What is a savitzky-golay filter," *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

Semantic Grid Map based LiDAR Localization in Highly Dynamic Urban Scenarios

Chenxi Yang[1], Lei He[1], Hanyang Zhuang[2], Chunxiang Wang[1], Ming Yang[12]

***Abstract*— Change-over-time objects such as pedestrians and vehicles remain challenging for scan-to-map pose estimation using 3D LiDAR in the field of autonomous driving because they lead to incorrect data association and structural occlusion. This paper proposes a novel semantic grid map (SGM) and corresponding algorithms to estimate the pose of observed scans in such scenarios to improve robustness and accuracy. The algorithms consist of a Gaussian mixture model (GMM) to initialize the pose, and a grid probability model to keep estimating the pose in real-time. We evaluate our algorithm thoroughly in two scenarios. The first scenario is an express road with heavy traffic to prove the performance towards dynamic interferences. The second scenario is a factory to confirm the compatibility. Experimental results show that the proposed method achieves higher accuracy and smoothness than mainstream methods, and is compatible with static environments.**

I. Introduction

3D-LiDAR-based pose estimation is one of the most widely used on-line vehicle self-localization methods in the Global Navigation Satellite System (GNSS) signal denied or disturbed environments for autonomous driving. Data association from the observed scans to the pre-defined environmental map (scan-to-map) is the most critical step for such approaches, and the association reliability mainly determines the system performance. Dynamic interference in the observed scans has been a long-term challenge for data association in two aspects: (1) It provides time-varying features not present in the map that lead to incorrect data association; (2) It occludes extensive environmental features that lead to data association quantity reduction.

In order to suppress such impacts, one approach is to perceive the presence of incorrect data association during the pose convergence iteration and thus eliminate dynamic objects. However, this approach relies on the fact that the correct data association takes the dominant effect so that the outliers can be identified based on their significant distance error. Therefore, it's time-consuming and relatively unreliable. In this study, we focus on another approach, which is to directly exclude dynamic objects from the observed scans before data association. Our idea is to ensure the stationary status of the data association candidates by introducing semantic features. Such features should be widespread and generally static in urban environments.

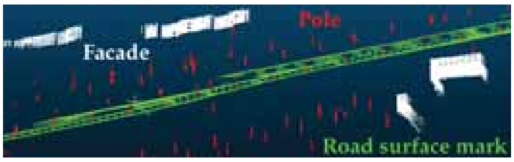


Fig. 1. The semantic point cloud map (SPCM) used in this study contains a sparse point cloud with only three kinds of static semantic features, which are poles, facades, and road surface marks.

Yu et al. [1] propose a semantic alignment method for city-scale LiDAR data by the data association limited to six kinds of semantic objects extracted from two dense point cloud maps. The features include facades, roads, poles, cars, segments, and lines. Their method achieves higher alignment accuracy than the mainstream methods. However, these semantic features (i.e., cars) are not designed to solve the challenges of dynamic scenarios, and a single scan is too sparse to extract segments and lines. Their map-to-map alignment method also faces a considerable gap to meet the real-time requirement of the vehicle self-localization. Inspired by their idea, we extract the static semantic features of poles, facades, and road surface marks using a dense semantic segmentation method [2]. Fig. 1 shows the semantic point cloud maps (SPCM) generated in this process.

Comparing to the inter-frame LiDAR-odometry, it is more accurate and robust to utilize the semantic information in real-time localization based on pre-defined maps with global consistency. However, there are several unique challenges remaining for the scan-to-map localization of this study: (1) The relative structural loss of the observed scan comparing with the pre-defined map is much more significant than that in inter-frame scans of the LiDAR-odometry; (2) Computing efficiency is highly required to reach real-time performance; (3) The pose initialization needs to be fast enough to complete initial localization in large scale maps.

In this study, we propose a novel semantic grid map (SGM) based on the SPCM, in order to improve the scan-to-map localization by alleviating the aforementioned three challenges. The semantic categories and the corresponding probabilities are assigned to each grid. By projecting the SPCM into SGM, we significantly speed up the calculation while guaranteeing the robustness towards dynamic interference. We realize a Gaussian mixture model (GMM) to initialize the pose of the observed scan in the SGM. After the initialization, we design a grid probability model to keep track of the vehicle in the SGM. We evaluate our method on an express road with heavy traffic. In both the pose

This work is supported by the National Natural Science Foundation of China (U1764264/61873165).

[1] Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240; Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240 mingyang@sjtu.edu.cn.

[2] University of Michigan - Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai, 200240, China.

initialization and real-time localization, the proposed SGM and corresponding algorithms outperform the mainstream methods in precision and calculation speed. We also apply our method in a factory with generally static environment to confirm the compatibility.

The rest of the paper is organized as follows. Section II reviews related work. Section III introduces the SGM representation. Section IV describes our localization algorithms in detail. Section V provides the experimental evaluation of our method. Finally, Section VI concludes this paper.

II. Related Work

LiDAR localization approaches based on pre-defined maps can be classified according to their data association methods.

Point-based methods represented by Iterative Closest Points (ICP) [3] directly associate the observed points to the points on the map and get the result by converging the point-to-point optimization functions. Normal Distributions Transform (NDT) [4] transforms localization into a probability problem to solve and increases the robustness. Deschaud [5] proposed the IMLS-SLAM method to solve the data association problem by least-squares optimization and achieved high accuracy on the KITTI database. However, the point-based methods generally have poor real-time performances, therefore not suitable for on-board applications.

Gird-based methods can improve real-time performance by shrinking the map size. Levinson et al. [6] achieved a centimeter-level localization accuracy based on the probabilistic grid whereby every cell is represented as its own Gaussian distribution over remittance values. Wan et al. [7] proposed a similar method that is widely used in autonomous driving projects. Yang et al. [8] relieved ICP's local minima problem by combining a branch-and-bound (BnB) scheme. These methods can achieve high localization accuracy with great calculation speed in low dynamic scenes. However, just like the point-based methods, once the static correct data associations don't take the dominant effect, such methods will also suffer a severe accuracy loss.

Feature-based methods are currently the mainstream approach. Such methods extract abstract geometric structures such as lines and planes [9]. Generalized ICP (GICP) [10] realizes a plane-to-plane strategy that adopts the covariance matrices of the local surfaces to match with the point cloud. The Normal ICP (NICP) [11] assigns local geometric information of normal and curvature to the points to enrich the matching dimensions to improve the robustness further. LiDAR Odometry and Mapping (LOAM) [12], as one of the state-of-the-art methods, extracts edges and planes with great accuracy from the relatively sparse point clouds. Shan et al. [13] proposed a lightweight and ground-optimized LOAM variant that improved both speed and accuracy. Although these methods emphasize the structure of the objects, dynamic objects such as vehicles can also form local structures with strong consistency. Therefore, although feature-based methods are more robust to dynamic interference, they still cannot overcome the challenges brought by high-dynamic scenarios.

Descriptor-based methods cluster the point cloud into blocks and calculate the similarity between the observed scans and the maps based on the geometric measurement criteria. Dubé et al. [14] trained a similarity criteria. Lu et al. [15] designed a deep learning network to learn the point cloud characteristics and established the corresponding descriptor. Both these two methods efficiently improved the global localization performance, however, they have relatively slow calculation speed and poor interpretability.

To the best of our knowledge, the semantic category is one of the few (if not the only) features that can directly exclude the dynamic objects from the data association. Compared with the use of semantic cues in image-to-map registration tasks[16], LiDAR-based scan-to-map registration is more challenging due to the sparse information. Pole-like objects [17], [18], [19] and road surface marks [20], [21], [22] are often used for data association as their semantics strongly indicate these objects are static. In specific scenarios, these methods can eliminate the drawbacks caused by dynamic objects. But, relying on a single semantic feature will often result in a localization failure because of the occlusion and lack of structure. As introduced in the last section, Yu et al. [1] proposed a semantic alignment method that combined multiple semantic features to achieve higher localization accuracy. Parkison et al. [23] proposed a localization method based on the high-precision semantic segmentation of the dense point cloud. Chen et al. [24] computed semantic segmentation results in point-wise labels for the whole scan, allowing them to build a semantically-enriched map with labeled surfels. The global semantic segmentation process in these methods is time-consuming even on high-performance processors, therefore, it is almost impossible for on-line real-time applications.

III. Semantic Grid Map Representation

To accurately estimate the vehicle position, sufficient pose constraints from various directions and elevations are necessary. However, due to the sparseness of the point cloud, the static semantics extractable from a single LiDAR scan is relatively limited. For high-layer and ground semantics, facades and road surface marks are two robust static ones widespread in urban scenarios. However, in the middle layer, where the dynamic interferences are the most severe, it's typically challenging to find such features. Our idea is to strictly limit the static semantics, so as to distinguish them from the potentially dynamic ones effectively. Therefore, we choose only pole-like features, which implies that the objects are tree trunks or telephone poles.

Fig. 2 demonstrates the data structure of the proposed semantic grid map. Each grid is represented by the category determined by the semantic feature that has the most points, and the corresponding probability that is the proportion within the total points of this grid. Since the wrong data associations often occur at the boundaries of different categories (such as poles at the edges of the facade), the introduction of probability can weaken such impact. In some rare cases, SPCM contains some invalid semantic points, such as poles

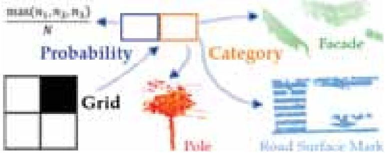


Fig. 2. Semantic grid map representation. Each occupied grid contains the information of the category and the corresponding probability.

formed by reeds. Although such points generally belong to static objects, it's obvious that they are not stable in the long term. What these features have in common is that their point clouds are more sparse than those intrinsically static features. Therefore, this paper removes such structures by a point number threshold.

IV. Localization

In this section, we describe our algorithm for the on-line pose initialization and real-time vehicle self-localization tasks. We denote the coordinate of the units in the sub-map M of the SGM as $m_1, ..., m_J$, and the units in the observed scan S as $s_1, ..., s_K$, where J and K are the number of units respectively. For the initialization task, the original pose must search a wide range to avoid various local minima. At the same time, the calculation can take relatively longer (several seconds is acceptable). Therefore, to keep as much map detail as possible, the SGM is in 3D formed by cubes. On the contrary, the localization task can inherit a much more accurate initial position from the previous frame while it requires strict real-time performance (typically 100ms), the SGM is in 2D formed by squares.

A. On-line Pose Initialization

In order to initialize the vehicle pose in GNSS denied areas, this paper proposes a GMM-based semantic categories to represent the pose initialization problem. We first generate a 3D SGM, and characterize each semantic category of this study as a Gaussian model. This model only focuses on the horizontal distribution as all of the three semantic categories in this study are vertically uniformly distributed. We can regard each cube of the observed scan $S_{K\times 3}$ as a mean value of the GMM, and each cube of the sub-map $M_{J\times 3}$ as the corresponding Gaussian distributed samples. The response probability of the GMM can be represented as

$$P(m_j) = \sum_{k=1}^{K} P(s_k) P(m_j|s_k) \tag{1}$$

where $P(s_k)$ is each component of the GMM of cube k. By considering the category probability of m_j and a penalty term for outliers and noise w inspired by [25], we can extend the expression as

$$\begin{aligned} P(m_j, C_{m_j}) &= w\frac{1}{J} \\ &+ (1-w)\sum_{k=1}^{K} P(C_{m_j}|m_j, s_k) P(m_j|s_k) P(s_k) \end{aligned} \tag{2}$$

where C_{m_j} is the corresponding semantic category of cube m_j. We define the semantic confidence to associate the k^{th} scan cube to the j^{th} map cube as

$$P(C_{m_j}|m_j, s_k) = \begin{cases} \frac{\max(n_p, n_f, n_r)}{N} & C_{m_j} = C_{s_k} \\ 0 & C_{m_j} \neq C_{s_k} \end{cases} \tag{3}$$

where C_{s_k} is the semantic category of cube s_k of the scan, and n_p, n_f, n_r are the number of points in each semantic category of poles, facades, and road surface marks of cube m_j respectively, while N is the total point number in cube m_j. And we have

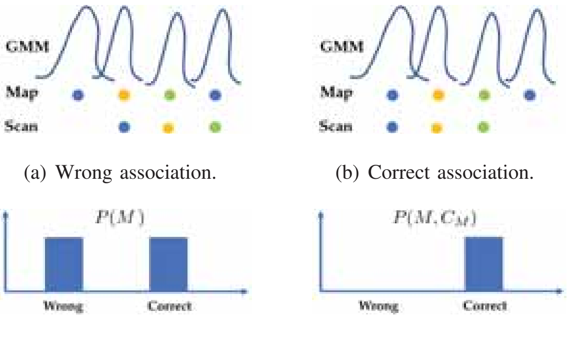
$$P(m_j|s_k) = \frac{1}{2\pi|\Sigma_k|^{\frac{1}{2}}} \exp(-\frac{1}{2}(m_j - s_k)^T \Sigma_i^{-1} (m_j - s_k)) \tag{4}$$

where Σ_k is the variance of the k^{th} component need to be solved. The pose initialization can be represented as

$$T^* = \arg\max_{T} P(M, C_M) = \prod_{j=1}^{J} P(m_j, C_{m_j}) \tag{5}$$

where the transformation matrix T is to decide the data association pairs of m_j and s_k in Equ. 2. T^* is to be found by maximizing the data association probability.

To solve the Σ_k and T, we use the expectation-maximization (EM) algorithm, whose solving process can be found in [25]. The role of semantic categories in this process is shown in Fig. 3. The three semantic categories are denoted as circles in blue, yellow, and green. Traditional non-semantic localization methods like CPD only consider the geometric distances between the points (or grids) between the observed scan and the map. Therefore they cannot distinguish the wrong data association (3(a)) and the correct one (3(b)).



(a) Wrong association.

(b) Correct association.

(c) Non-semantic probability.

(d) Semantic probability.

Fig. 3. An example of localization initialization using semantic categories in the probabilistic data association.

Considering the resolution difference between one frame LiDAR scan and the dense map, which is challenging for scan-to-map pose estimation, the semantic grid map representation can efficiently narrow such gap by down-sampling the map and enhancing the sparse scan at the same time. The GMM ensures such a strategy to reach a localization accuracy exceeding the grid resolution.

B. Real-time Localization

Probabilistic data association provides an effective framework for solving the impact of incorrect data association on the localization algorithm. As mentioned at the beginning of this section, we denote the squares in the 2D SGM as $M = \{m_j\}$, and the squares in the observed scan after gridding as $S = \{s_k\}$. The associated pairs set between M and S is denoted as $A = \{a_{j,k}\}$ where $a_{j,k} = (m_j, s_k)$. The residual error is denoted as $\varsigma = M - T \times S$ where T is the transformation matrix. The semantic category is denoted as C. The localization problem can be represented as

$$T^* = \arg\max_T P(\varsigma, C, A|M, S) \tag{6}$$

Use the Bayes Rule, this product is factored as

$$\begin{aligned} P(\varsigma,&C, A|M, S) \\ &\propto \underbrace{P(\varsigma|A, M, S)}_{error} \underbrace{P(C|A, M, S)}_{label} \underbrace{P(A|M, S)}_{geometry} \end{aligned} \tag{7}$$

The error term is defined as

$$P(\varsigma|A, M, S) = \prod \exp(\frac{-\|m_j - Ts_k\|^2}{2}) \tag{8}$$

and the label term is same to Equ. 3.

Eventually, this paper adopts the traditional geometric association as a protection term. To avoid overemphasizing the effect of Euclidean registration and thus weakening the semantic information, this paper uses the k nearest neighbors (KNN) structure under the uniform distribution to facilitate the search of the k nearest association category as

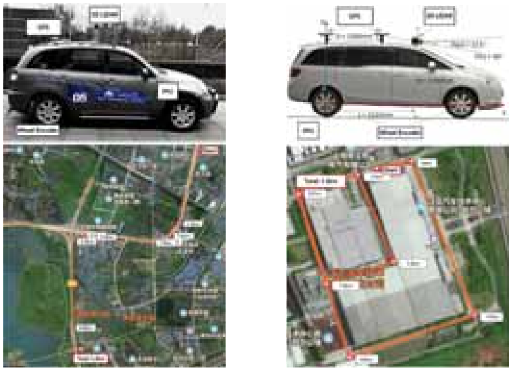
$$p(A|M, S) = \begin{cases} 1/k & knn \\ 0 & otherwise \end{cases} \tag{9}$$

According to the above association method, this paper assumes that the errors conform to the Gaussian distribution. The model needs to solve two unknown variables, one is data association probability, and the other is the pose transformation matrix T. The EM algorithm is also used to solve this problem as elucidated in reference [25].

V. Experimental Evaluations

We evaluated our method in two scenarios. The first one is an express road with heavy traffic to test the performance under strong dynamic interferences, as shown in Fig. 4(a). The second one is a factory with a generally static environment to confirm the compatibility, as shown in Fig. 4(b). The two vehicle platforms are equipped with a HESAI Pandar-40P LiDAR and a Velodyne VLP-16 LiDAR respectively. Both platforms have computing resources of the Intel i7-7567U CPU @3.5GHz with 16GB memory. The calculation times of our method in the experiments include the semantic features extraction from the observed scan, which is based on geometric rules.

In the map generation process, the GNSS positioning results are used as ground truth data. Then, the SPCM is generated from the semantically segmented point cloud consistent with GNSS [2]. Fig. 5 shows a part of the SPCM of the express road with the three semantic categories.



(a) Express road. (b) Factory.

Fig. 4. The experimental platforms and environments.

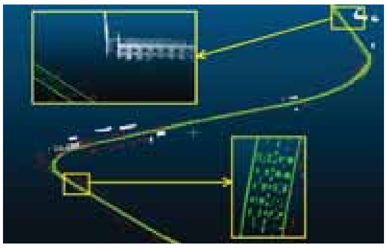


Fig. 5. An example of SPCM of the express road. The three semantic categories (poles-red, facades-white, road surface marks-green) can be easily identified.

A. On-line Pose Initialization

We randomly selected 100 different poses on the straight express road to evaluate our method. The SGM is a 3D grid map that each grid is a cube with a side length of 0.2m. As the pose initialization is sensitive to both the horizontal offset and the orientation error, in this experiment the horizontal offset is set as a uniform distribution up to 50m, and the orientation error is set to 30°, 60°, and 90°.

We compared our results with Coherent Point Drift (CPD) [25], which is a widely used method for pose initialization. The result can be found in Table I. It shows that our method has a better robustness and overall accuracy especially when the initial pose is set with the orientation error over 60°. We also compared the calculation time of orientation error at 90°which is considered as the worst case. The proposed method takes less than half time than CPD. It proves that the semantic-category-based method in this paper can speed up the iteration and reduce the time consumption.

From Table I we can see that CPD generally converge to the correct heading angle when the orientation error is set to 90°on the straight road experiment. We also demonstrate a special case of the pose initialization at a conjunction of the express road to further illustrate the effectiveness using semantic categories under the same orientation setting, as shown in Fig. 6. The semantic categories of road surface marks and poles are represented in red and green, and the observed scan is rendered in white, as shown in Fig. 6(b). Because the distance between the road surface marks of the

TABLE I
ACCURACY AND CALCULATION TIME EVALUATIONS FOR THE POSE INITIALIZATION EXPERIMENT.

	Trans.(m)	30°	60°	90°
CPD	Mean	0.17	0.19	6.42
	Max	0.18	0.50	67.3
Our method	Mean	0.08	0.18	**0.13**
	Max	0.12	0.24	**0.30**
	Yaw.(°)	30°	60°	90°
CPD	Mean	0.18	0.19	3.83
	Max	0.20	0.48	6.80
Our method	Mean	0.13	0.13	**0.11**
	Max	0.16	0.16	**0.18**
	Calculation time (s)			90°
CPD	Mean			7.25
Our method	Mean			**3.23**

TABLE II
ACCURACY EVALUATIONS FOR THE LOCALIZATION EXPERIMENT ON THE EXPRESS ROAD.

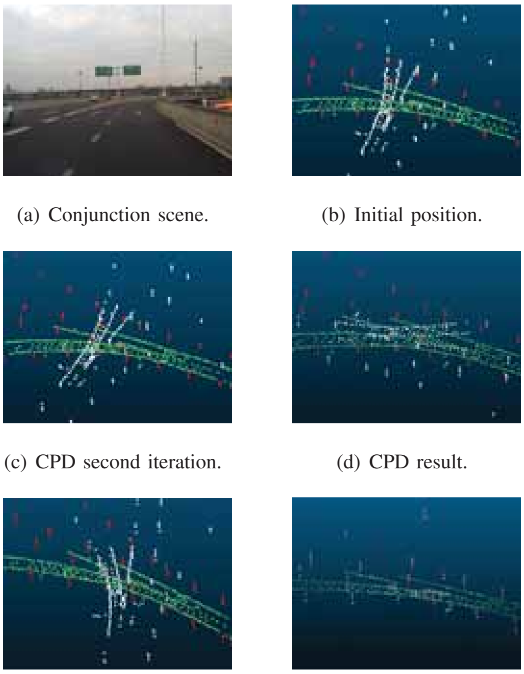
	Lat.(m)	Lon.(m)	Trans.(m)	Yaw.(°)
Semantic ICP	0.20	0.24	0.31	**0.20**
Grid Localization	0.11	≥ 2	≥ 2	≥ 2
Poles	0.37	0.33	0.55	1.86
Road marks	0.10	≥ 2	≥ 2	0.37
Facades	0.09	-	-	0.54
Our method	**0.08**	**0.12**	**0.16**	0.27

observed scan and the poles of the map are geometrically closer, CPD rotates to the wrong direction from the very beginning of the iteration, and eventually converged to the local minimum. On the contrary, the semantic category plays an important role, and leads the iteration to the correct pose.

B. Real-time Localization

In the 5.2km express road experiment, we compared our result with the Semantic ICP [23] and a traditional non-semantic approach of the occupancy grid localization using weighted point cloud[26]. We also compared our result with using each one of the three semantic categories separately to show the effectiveness using multiple semantic features.

Table II shows that our method significantly outperformed other methods and the stand-alone semantic categories in terms of transformation accuracy. For the Grid Localization method and stand-alone road surface marks, the express road is not a geometrically salient scenario in longitude (also known as corridor effect). Due to such failure, Gird Localization also failed to achieve reasonable yawing accuracy. For the same reason, the facades are parallel to the road direction; therefore, they can't provide any longitudinal pose constrain. The comparison of the calculation time proves the efficiency of this approach, as shown in Table III. Table IV compares the size of different kind of maps, from where we can see proposed semantic grid map takes up the smallest storage space.



(a) Conjunction scene.

(b) Initial position.

(c) CPD second iteration.

(d) CPD result.

(e) Our method second iteration.

(f) Our method result.

Fig. 6. A special case at the conjunction of the express road that CPD failed to converge to the correct position.

TABLE III
CALCULATION TIME EVALUATIONS FOR THE LOCALIZATION EXPERIMENT ON THE EXPRESS ROAD.

Method	Mean operation time(ms)
Semantic ICP	150.40
Grid Localization	44.03
Poles	15.14
Road marks	16.34
Facades	14.08
Our method	23.41

In the factory experiment, our method can also achieve a comparable accuracy to the express road condition, where the comparison can be found in Table V. The main reason is that in factory experiment, there are more structural features which are beneficial for SGM-based localization.

VI. CONCLUSION

In this paper, we proposed a localization method based on the semantic grid map (SGM) with poles, facades, and road surface marks. Such map is small in size and rich in information. By introducing the Gaussian mixture model (GMM) to the semantic features, the corresponding pose initialization method improved the robustness and accuracy while reduced the calculation time by half comparing to the traditional non-semantic baseline. In the real-time localization process, this

TABLE IV

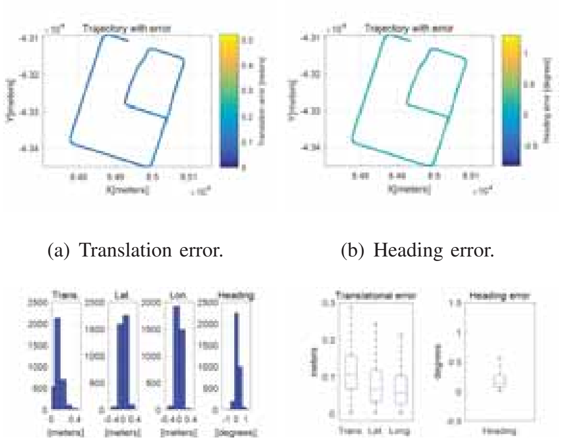
MAP STORAGE SIZE COMPARISON.

Map structure	Size(MB/km)
Point cloud map	$\geq$ 1000
Semantic point cloud map	34
Grid map	5.3
Semantic grid map (Ours)	**1.1**

TABLE V

ACCURACY COMPARISON BETWEEN THE DYNAMIC SCENES AND STATIC SCENES FOR THE LOCALIZATION EXPERIMENT.

	Trans.(m)	Lat.(m)	Lon.(m)	Yaw.(°)
Express road	0.14	0.06	0.11	0.21
Factory	0.12	0.07	0.08	0.19



(a) Translation error.

(b) Heading error.

(c) Error histogram.

(d) Error boxplot.

Fig. 7. Localization result in the factory environment.

paper introduced grid probability to implement a new data association strategy with semantic information. Experimental results show that our proposed method is robust and accurate in not only dynamic scenarios, but also static environments which guaranteed the adaptability.

REFERENCES

[1] F. Yu, J. Xiao, and T. Funkhouser, "Semantic alignment of lidar data at city scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, MA, United states: IEEE, 2015, pp. 1722–1731.

[2] Y. Chen, M. Yang, C. Wang, and B. Wang, "3d semantic modelling with label correction for extensive outdoor scene," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1262–1267.

[3] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[4] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro universitet, Stockholm, 2009.

[5] J.-E. Deschaud, "Imls-slam: Scan-to-model matching based on 3d data," in *2018 IEEE International Conference on Robotics and Automation(ICRA)*. Brisbane, QLD, Australia: IEEE, 05 2018, pp. 2480–2485.

[6] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4372–4378.

[7] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD, Australia: IEEE, 2018, pp. 4670–4677.

[8] J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1457–1464.

[9] F. Pomerleau, F. Colas, R. Siegwart *et al.*, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends® in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[10] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[11] J. Serafin and G. Grisetti, "Nicp: Dense normal based point cloud registration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 742–749.

[12] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[13] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain: IEEE, Oct 2018, pp. 4758–4765.

[14] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, 2017, pp. 5266–5272.

[15] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-net: Towards learning based lidar localization for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, USA: IEEE, 2019, pp. 6389–6398.

[16] D. P. Paudel, A. Habed, and L. Van Gool, "Optimal transformation estimation with semantic cues," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 4668–4677.

[17] R. Spangenberg, D. Goehring, and R. Rojas, "Pole-based localization for autonomous vehicles in urban scenarios," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, Korea: IEEE, 2016, pp. 2161–2166.

[18] M. Sefati, M. Daum, B. Sondermann, K. D. Kreisköther, and A. Kampker, "Improving vehicle localization using semantic and pole-like landmarks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. Redondo Beach, CA, United states: IEEE, 2017, pp. 13–19.

[19] A. Kampker, J. Hatzenbuehler, L. Klein, M. Sefati, K. D. Kreiskoether, and D. Gert, "Concept study for vehicle self-localization using neural networks for detection of pole-like landmarks," in *International Conference on Intelligent Autonomous Systems*. Baden-Baden, Germany: Springer, 2018, pp. 689–705.

[20] F. Poggenhans, N. Salscheider, and C. Stiller, "Precise localization in high-definition road maps for urban regions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain: IEEE, 10 2018, pp. 2167–2174.

[21] A. Hata and D. Wolf, "Road marking detection using lidar reflective intensity data and its application to vehicle localization," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Qingdao, China: IEEE, 2014, pp. 584–589.

[22] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based low-cost vehicle localization system for complex urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1078–1086, 2016.

[23] S. A. Parkison, L. Gan, M. G. Jadidi, and R. M. Eustice, "Semantic iterative closest point through expectation-maximization." in *BMVC*, 2018, p. 280.

[24] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4530–4537.

[25] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[26] L. Guo, M. Yang, B. Wang, and C. Wang, "Occupancy grid based urban localization using weighted point cloud," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Rio de Janeiro, Brazil: IEEE, 2016, pp. 60–65.