Introduction to Connected and Autonomous Driving

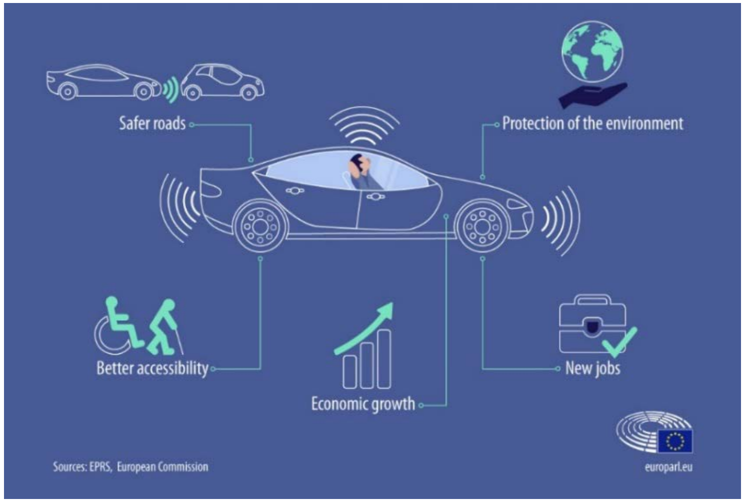
Cheng-Zhong Xu
FST & IoTSC
czxu@um.edu.mo

Outline

- Autonomous Driving
- AD History and Current Status
- Key Enabling Technologies
  - Sensor and Fusion
  - Environment Map
  - Environment Perception
  - Motion Planning
  - Control and Actuation
- V2X Comm and Edge AI for Connected Vehicles
- MoCAD: Macao Connected and Autonomous Driving
- Short Course Overview

Autonomous Driving

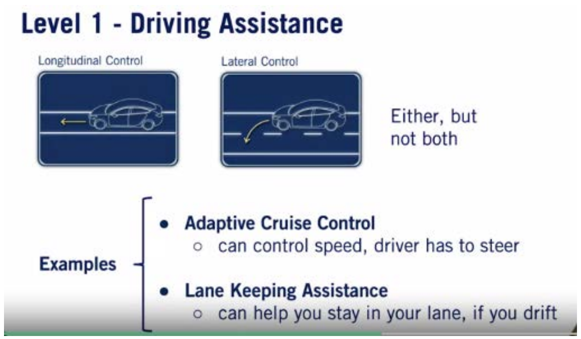
- **Autonomous vehicle (AV),** is a vehicle that is capable of sensing its environment and moving safely, economically, and comfortably with little or no human input.
  - Self Driving Car
  - Driverless Car
  - Automated Driver Assistant System (ADAS)

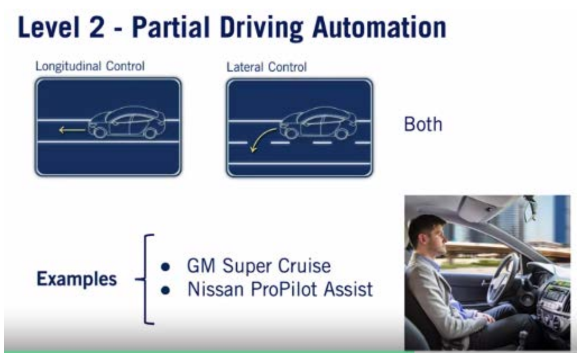


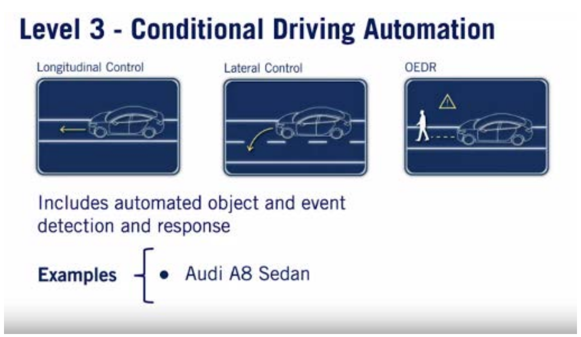
- Societal and economic benefits: inevitable trend of auto industry and change of the way of living and moving
  - Safe Road: 1.2M/year death to ZERO fatal accidents (94% due to human errors)
  - Better accessibility: ZERO waiting time (before and during travel)
  - Increase efficiency: ZERO emission
  - Disruptive techs reshape job market, push for economic growth

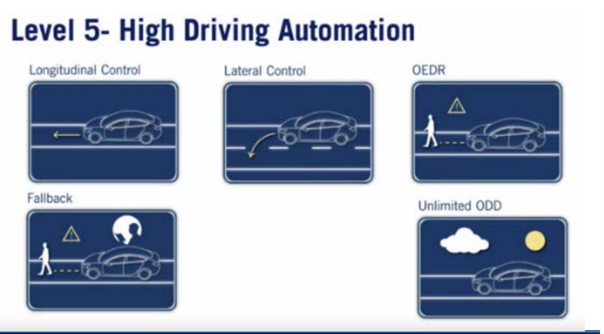
Levels of Autonomy (SAE)

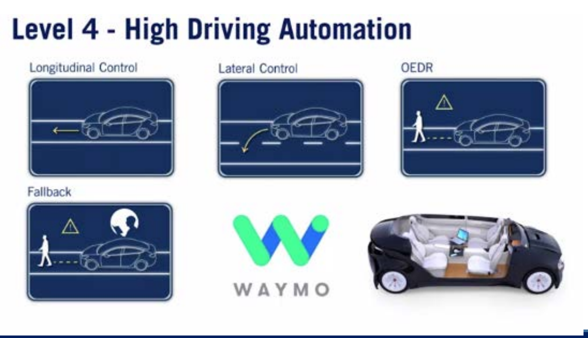
- Level 0: Regular driving, w/o autonomy. Warning only.
- Level1 (hands on): Either Lateral or Longitudinal Control
  - Adaptive Cruise Control
  - Lane Keep Assistance, parking assistance
- Level 2 (hands off): Both Longitudinal and Lateral Control
  - Tesla Model-3, GM Super Cruise, Nissan ProPilot Assist
- Level 3 (eyes off): Automated Obj-Event Detection and Response (OEDR)
  - Audi A8 Sedan
- Level 4 (mind off): Handle emergency autonomously, Restricted Operational Design Domain (ODD). Fallback when automation fails
  - Waymo (google)
- Level 5: Unlimited ODD











Autonomous Vehicle Technologies

- **AV** is a complex intelligent control system, more than an mechanical vehicle body: sensing, comm and computing, control, AI tech
  - Sensors to perceive their surroundings
  - AI to interpret sensory info, develop understanding of scenes, and make decisions or recomm for moving and behaviors
  - Advanced control systems to convert the decisions to vehicle lateral, longitudinal and steering control in maneuvers
  - Internal comm (e.g. CAN) to integrate components together; external wireless comm helps deeper sensing to env, and secures assistance from cloud/edge
  - Each AV must be equipped with a central processing unit for management of the resources, and orchestration of their execution.
- Intelligent Driving (ID) is a way of autonomous driving, with assistance of AI technologies.

History

- Pre-history
  - 1925, Francis Udina, remote control car in the streets of Manhattan, NY
  - 1956, GM promoted a self-driving vehicle video, and expected to come true in 1976
- History (1986-1997)
  - 1986, Ernst Dickens from Univ of Munich, robotic van that drove fully autonomously without traffic, at 60 km/h (based on Transputer); 1990, drove from Munich to Copenhagen (1600km) with a mean distance between human intervention of 9 km.
  - At the same time, 1986, CMU's NavLab1 30km/h; 1990, NavLab2, 110 km/h; 1996, for across American tour (4800 km) with 98.2% to autonomous driving
  - 1997, UC Berkely demonstrated connected autonomous vehicles

[The story of Autonomous Vehicles]

History

- History (2004-2016)
  - 2004 DARPA Grand Challenge: none of them finished, but navigated 142 miles through a Desert
  - 2005 2nd Grand Challenge: 5/23 teams finished the test;
    Stanford Stanley first, seconded by CMU's Sandstorm
  - 2007, DARPA Urban Challenge, CMU first, Stanford 2nd
  - Right after, Google hired both teams and pushed their design onto public roads; In 2010, Google car logged over 140K miles in CA; 10 million of roads by Oct 2018
  - In 2012, Nevada issued the first testing license; As of 2019, twenty-nine U.S. states have passed laws permitting autonomous cars.
  - 2013,Europea, Japan
  - At the end of 2015, Tesla Autopilot;
  - In 2016, Taxi company startups: Zoox and nuTonomy
    (first operational in Singapore in 2016)

Stanley

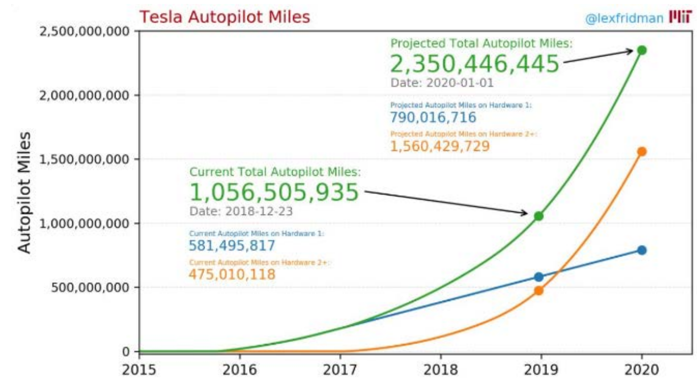
Tesla Autopilot

- In china
  - In Dec 2015, Baidu launched its apollo self-driving cars
    for both highway and urban envir (at a max rate of 100 km/h)
  - Baidu Robotaxi in 4.2020

Baidu Apollo

Waymo and Telsa

- Google Auto: Road testing: 24 million km since 2009
  - 5m km/day in controlled lab and simulation env.
- Telsa, 75m km for 6 mon, and increase 1m km/day. Totally, 24 billion km till 2019.



Tesla Autopilot Reaches 1 Billion Miles by 2018

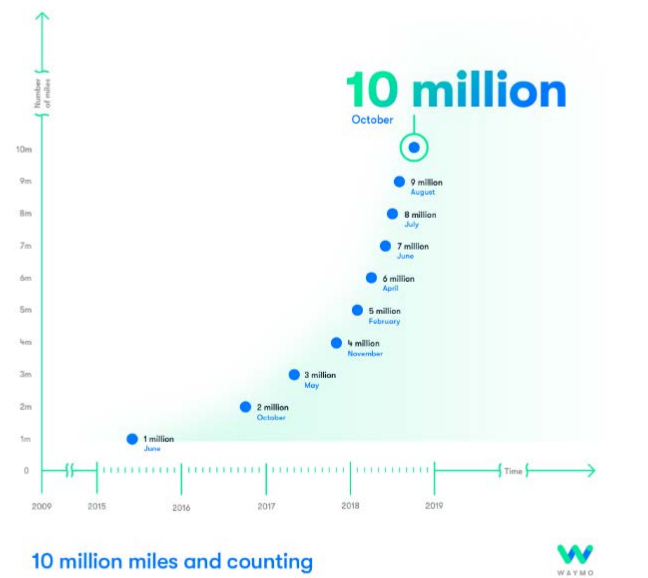
On-road tests of self-driving cars in California

(During the year until Nov. 2019)

Ranking *(Previous year)*	Testers'name *(Country)*	Distance covered *(In kilometers)*
1 (1)	Waymo *(U.S.)*	2,340,201
2 (2)	GM Cruise *(U.S.)*	1,337,426
3 (9)	PONY.AI *(China)*	281,386
4 (8)	Baidu *(China)*	174,292
5 (6)	Nuro *(U.S.)*	110,661

Source: State of California

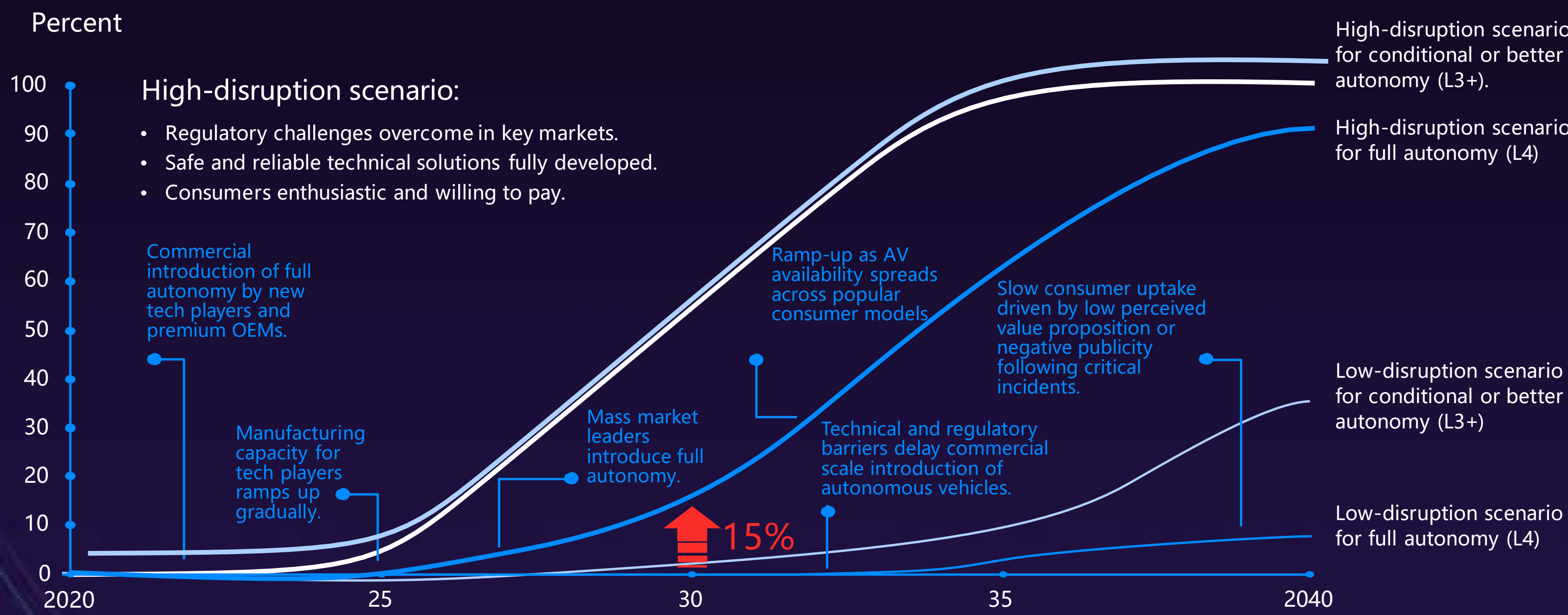
NIKKEI ASIAN REVIEW



Waymo Reaches 10 Million Miles in 2018

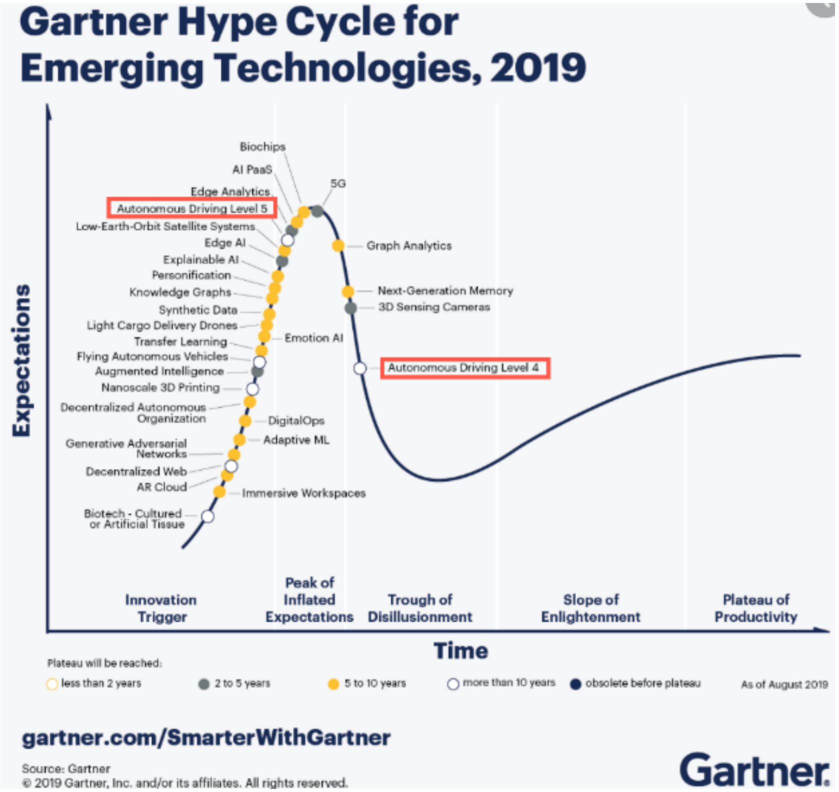
New vehicle market share of fully autonomous vehicles

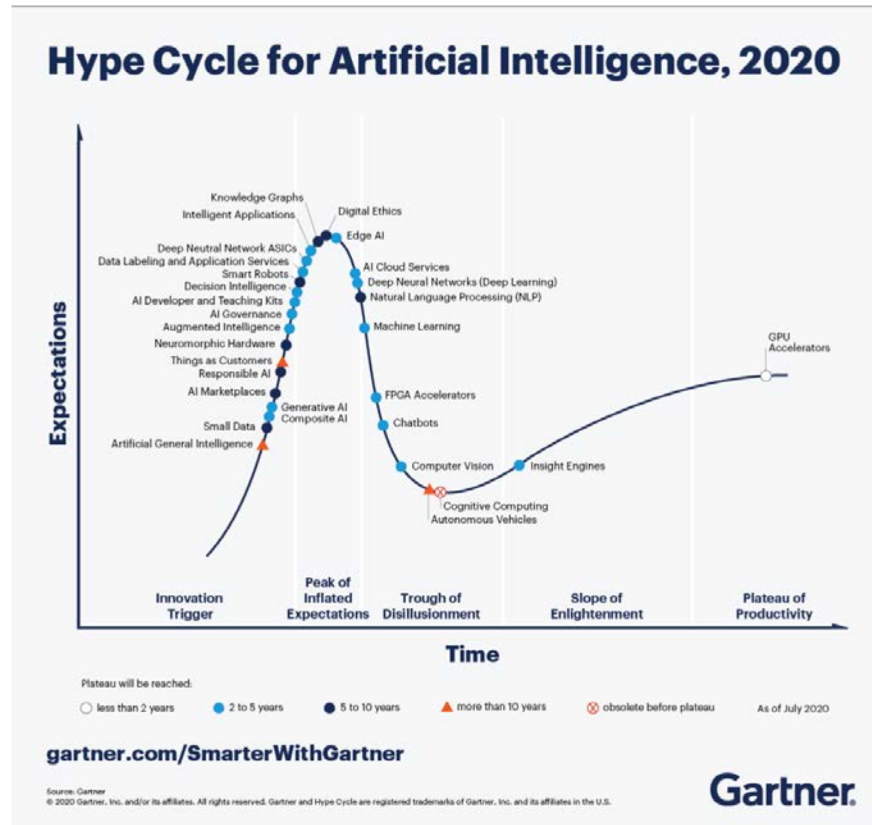
15% L4+ and 50% L3+ in 2030



Source: McKinsey-Baidu 2017

Market Prediction





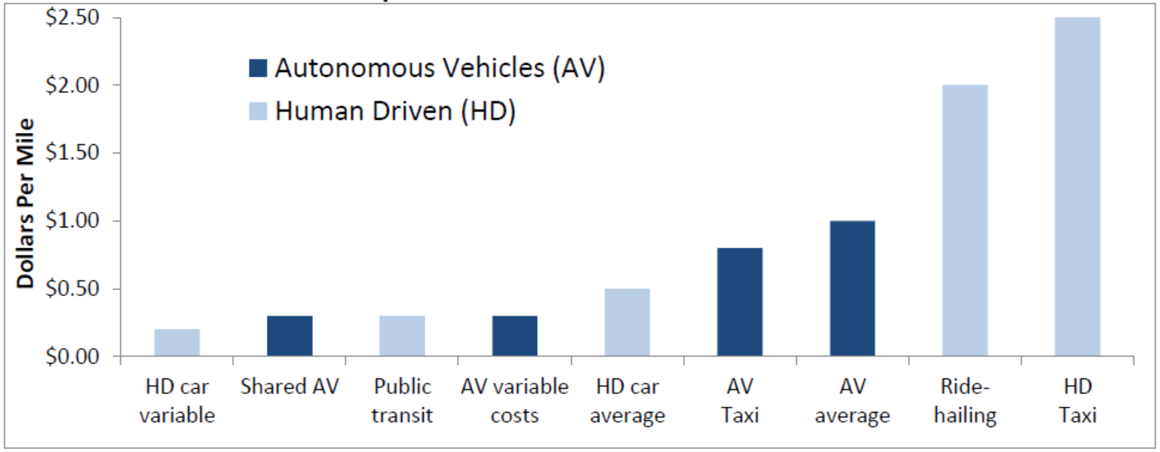
Rush to Market

Robotaxi for ride-sharing (driverless taxi)

Small-scale, often low-speed, almost always with safety driver

- NuTonomy (spinoff from MIT in 8.2016 in Singapore)
- Uber (9.2016 in Pittsburg, extended to SF later),
- Waymo (2017 in Phoenix), GM, Ford, Tesla,
- Baidu(Robotaxi, 4.2020),
- AutoX (8.2020 in Shanghai by Didi),
- etc

Exhibit ES-1 **Cost Comparison**



Autonomous vehicles (AVs) are likely to cost more than human-driven private vehicles (HVs) and public transit, but less than human-driven taxis and ridehailing services.

Source: Autonomous Vehicle Implementation Predictions, June 2020

Setbacks…

- In 2016, first fatal accident in Florida by Tesla (level 2). In March 2018, another crash, hitting into the barrier and killed the driver
- March 2018 Uber crash in temple Arizona, pedestrian fatality; Dec 2018 returned to testing (Level 3)
- Oct 2018, Google Waymo Van hit by a car in Arizona

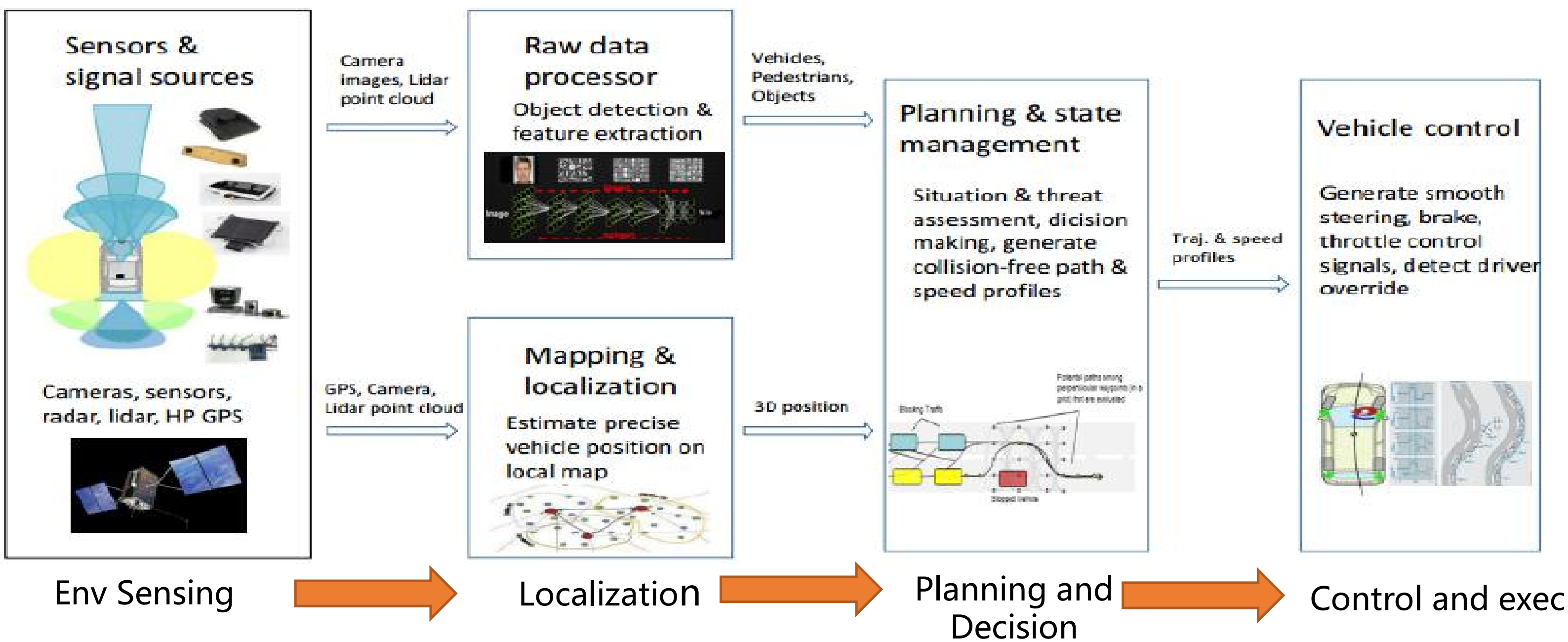
When will autonomy happen in pilot testing (with 10K+) under what conditions?

MIT Lex Fridman's Prediction:

- Autonomy will not be adopted because it is safer.
- Autonomy will not be adopted because it is faster.
- Autonomy will not be adopted because it is cheaper.
- Autonomy will be adopted if creates a better human experience.

Autonomous Driving Overview

- Driving mission: to navigate from point A to point B on the map
  - Output: safe, efficient, and comfortable planned path towards the goal
- Use sensors to perceive the surrounding environment (static obj and traffic signs, and moving objects), control the steering and speed of the vehicle, and perform dynamic path planning based on real-time road conditions



Sensor stream to control actuation

Key Enabling Technologies

- Sensor and Sensor Fusion
- Environment Map and Localization
  - High-Definition Map to be created offline, and updated online: LiDAR + Camera
  - Localization needs to be performed at centimeter order
- Perception of Environment
  - Obstacle detection and tracking: static object vs dynamic object
    - Deformable Part Models (DPM) algorithm to detect vehicls and pedestrains (based on image)
    - Use point-cloud data to detect objects by Euclidean clustering.
  - Scene understanding
    - Object Detection and Tracking, based on image vs on Point Cloud
    - 3D Scene Understanding
- Motion Planning
  - Long vs short term vs immediate plan
  - Reactive plan vs Predictive plan

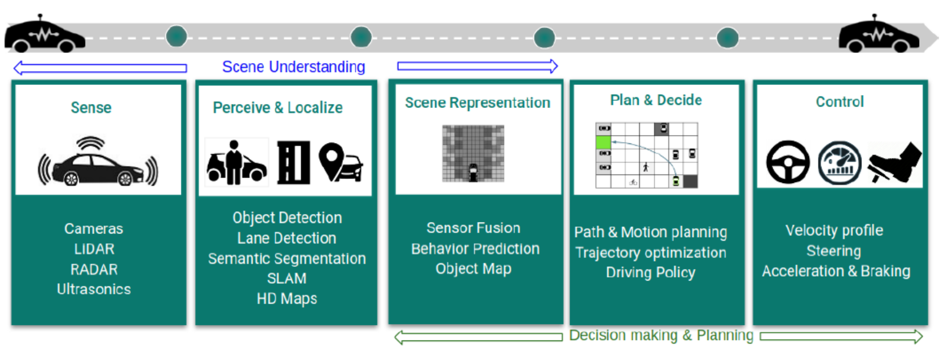


Fig. 1. Standard components in a modern autonomous driving systems pipeline listing the various tasks. The key problems addressed by these modules are Scene Understanding, Decision and Planning.

Sensor Features

- Cameras:
  - Visible light camera for detecting and recognizing objects, so the image data can be fed to AI-bed algorithms for object classification
  - Infrared camera: offer superior perf in darkness
  - - generate huge amount of video data to process
  - - Limited capabilities with visible light camera
  - Could be distorted
  - E.g. Mobileye, Tesla

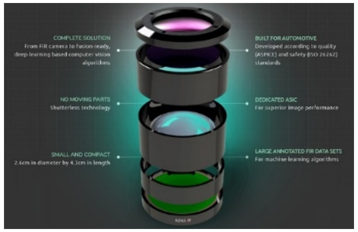
ASPECT 360° Panoramic Camera

- Radar: use radio wave to detect objects
  - Widely used for adaptive cruise control, e.g.
  - Short range radar operated at 24Ghz, Long range radar at 77GHz
  - Better to detect objects, accurately measure the distance
  - Limited ability to classify objects, poor vision abilities

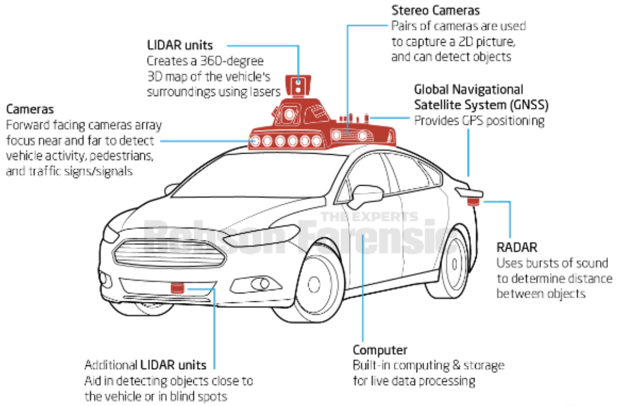
Navtech Radar 360° CIR Sensor

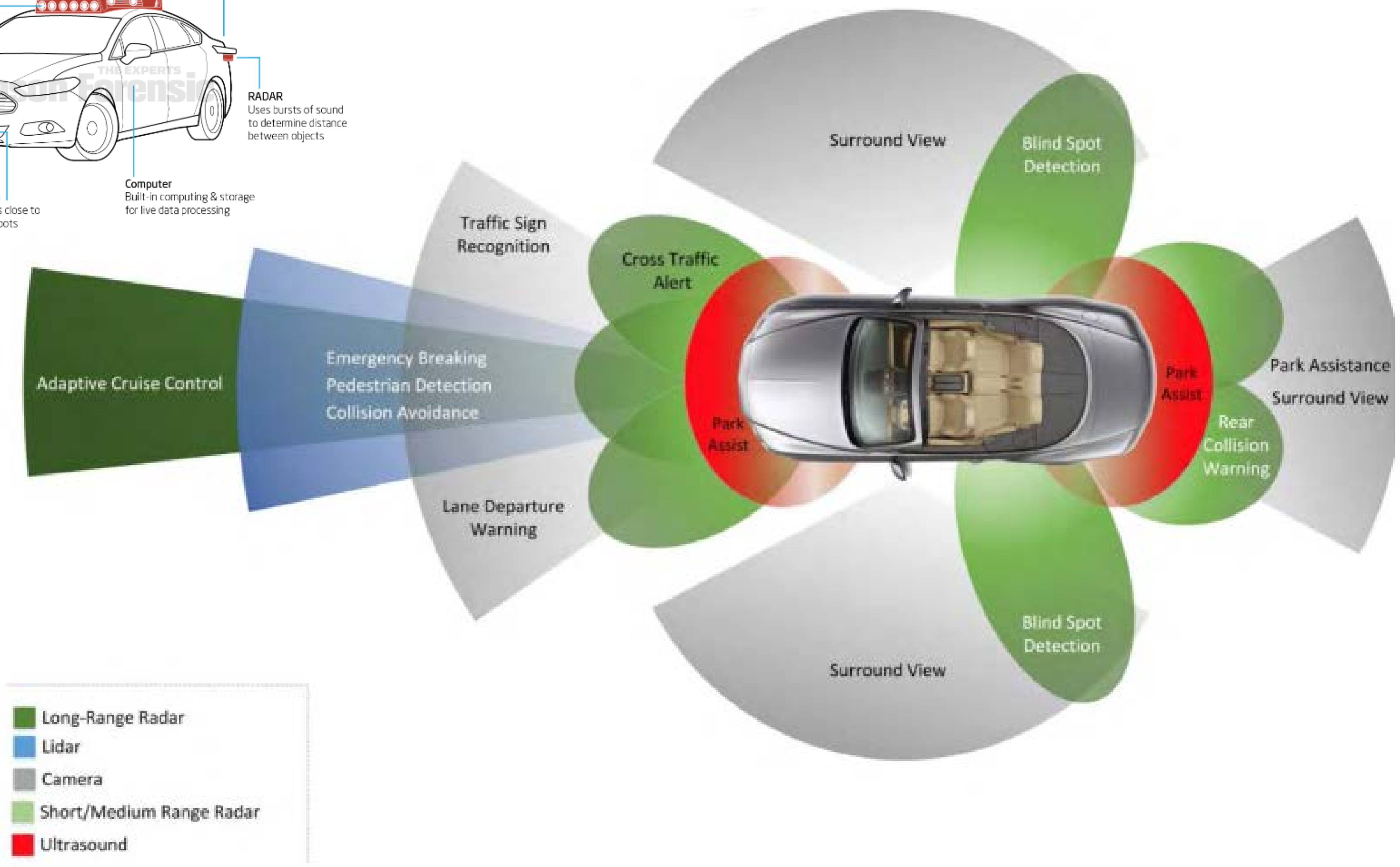
- LiDAR: use light waves
  - Provide 3D point-cloud data for localization and mapping; also used for measure the distance to surrounding objs.
  - Because of shorter wavelength, more accurate than Radar
  - Detection of Size and shape, in additional to distance.
  - Inability to cope with poor weather; too costly

- Ultrasonic sensor: for parking
  - Short range to a few meters
  - Support low-speed use cases
  - Distance!!
- GNSS/IMU: GNSS for absolute localization and IMU for 3D pose of the vehicle in space
- Far Infrared (FIR) thermal sensor, giving vehicles complete reliable detection of the road and its surroundings (Israel-based AdaSky)



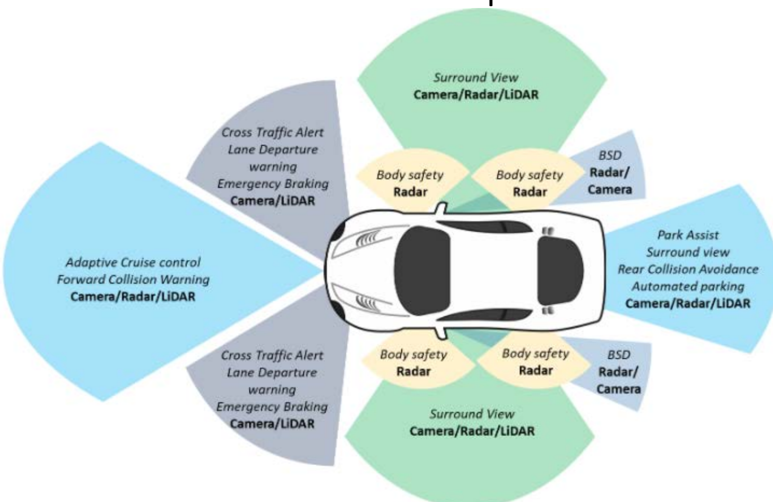
Sensor Features and Deployment

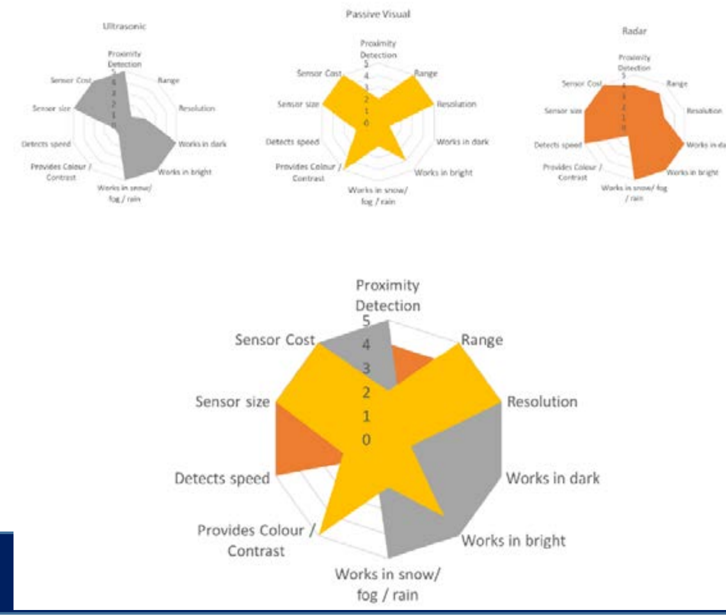




Sensor Fusion

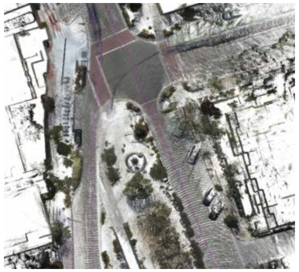
- Sensor fusion is combining of sensory data or data derived from disparate sources to reduce uncertainty of sensored info: sensors, soft sensors, history values of sensor data, etc.
  - Data level - to fuse raw data from multiple sources and represent the fusion technique at the lowest level of abstraction.
  - Feature level - features represent information computed on board by each sensing node. These features are then sent to a fusion node to feed the fusion algorithm. This procedure generates smaller information spaces with respect to the data level fusion, and this is better in terms of computational load.
- LiDAR vs Camera vs Radar
  - Radar info is too thin to differentiate objs of same size, same speed
  - Cameras see rich semantics. What you see is what you get. But no depth
  - LiDAR unable to deal with poor weather

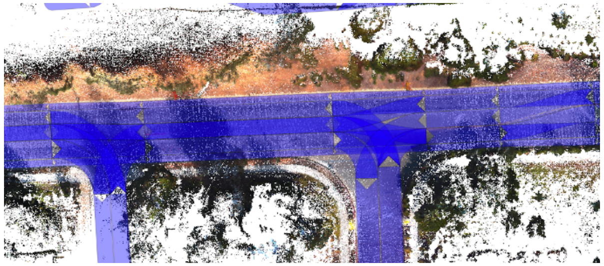


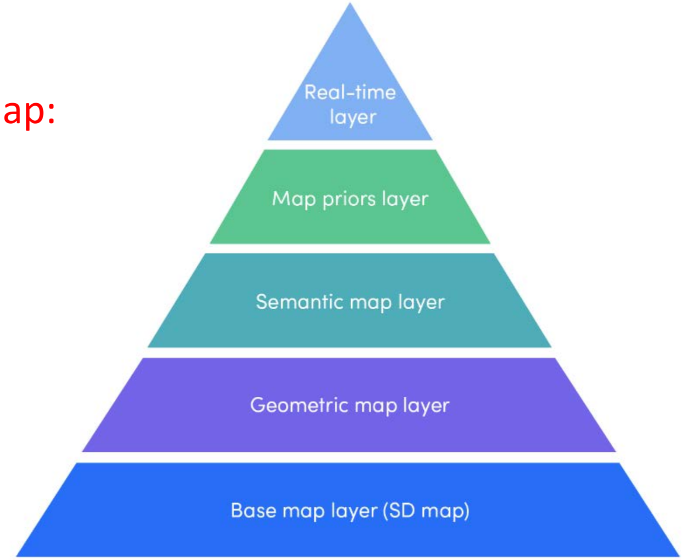


Environment Map

- A map of all static objects surrounding, created offline and updated online
- Hierarchical Structure:
  - Geometric map for 3D info of the world: use raw data of lidar, various cameras, GPS, and IMU are processed using SLAM alg to build a 3D view
  - Semantic map: semantic objects include 2D/3D traffic objects like lane boundaries, intersetions, crosswalks, parking spots, stop signs, taffic lights, etc
  - Map Prior: derived info about dynamic elements and human driving behaviors
  - Real-time traffic info: observed speeds, congestion, newly discovered construction zones, etc
- Map representation
  - Represented as a set of grid cells with probability
  - Occupancy Grid Map, Localization Map, Detailed Road Map:

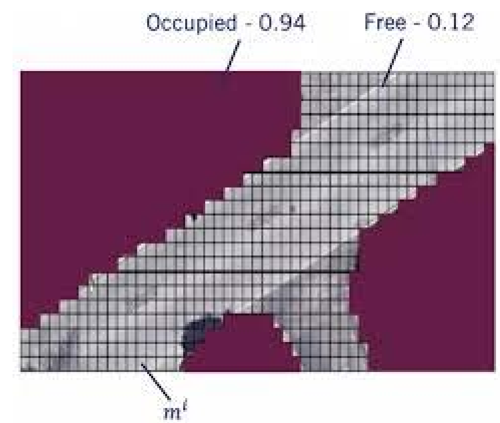


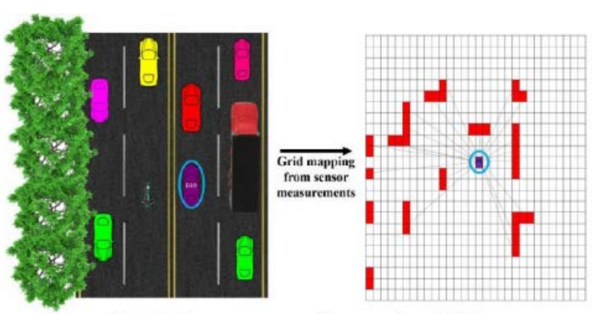




Occupancy Grid Map

- Grid-based representation to trivially incorporate info over multiple sensors and timesteps.
  - Noise inherent to lidar data used to construct occupancy grid
  - Creating accurate occupancy grid with noisy data by using Bayesian updates
- Grid-based discrete random field structure makes it highly agreeable to processing by CNN, allowing to take advantage of powerful developments from deep learning
  - Semantic occupancy grid encodes the presence or absence of an object category at each location
- Sensor fusion for occupancy grid: camera for high-level 2D info such as color, intensity, density, an and edge info. Lidar provides 3D point cloud data





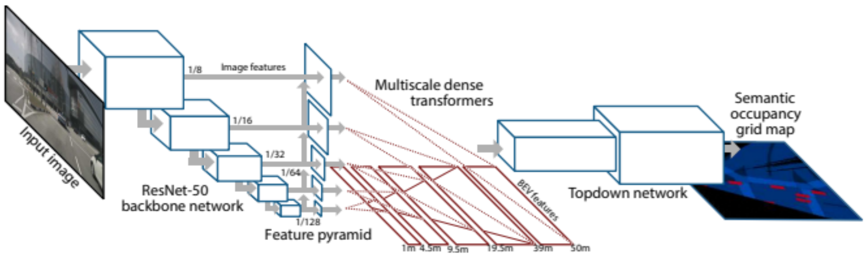
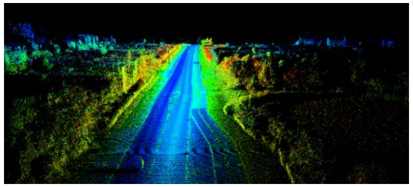
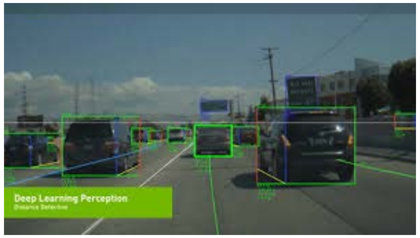


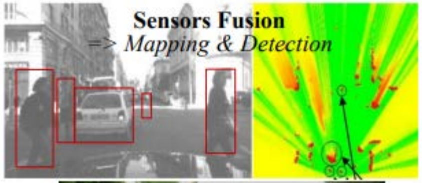
Figure 2. Architecture diagram showing an overview of our approach. (1) A ResNet-50 backbone network extracts image features at multiple resolutions. (2) A feature pyramid augments the high-resolution features with spatial context from lower pyramid layers. (3) A stack of dense transformer layers map the image-based features into the birds-eye-view. (4) The topdown network processes the birds-eye-view features and predicts the final semantic occupancy probabilities.

Environment Perception

- To understand the world around the vehicle in real-time, by using sensors, sensor fusion, SW systems, V2X components, etc
  - To detect other cars or pedestrians sharing the road
  - To know their unknowns: reason about visibility (e.g. 3D data frm lidar to represent objects as a point cloud and then try to match those point clouds to a library of 2D representations of objects)
- Perception:
  - Visual perception
  - Pedestrian perception
  - Traffic light/road signs perception
  - Traffic/congestion perception (drive w/o traffic, drive w. traffic)
  - Security/privacy perception, risk perception
- Challenges:
  - Noisy data, incompleteness, dynamicity, discrete measurements
  - Strong embedded & realtime constraints
- Deep learning for scene understanding
- Approach: embedded Bayesian perception
  - Reasoning about uncertainty & time window
  - Improving robustness using Bayesian Sensor Fusion
  - Interpreting the dynamic scene using Contextual and Semantic info
  - SW/HW integration using GPU, multicore, …

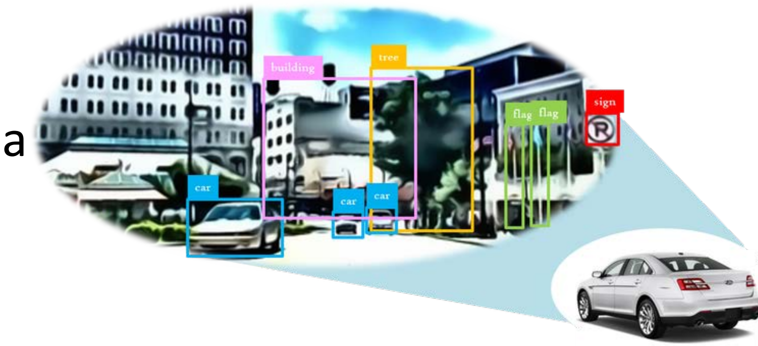




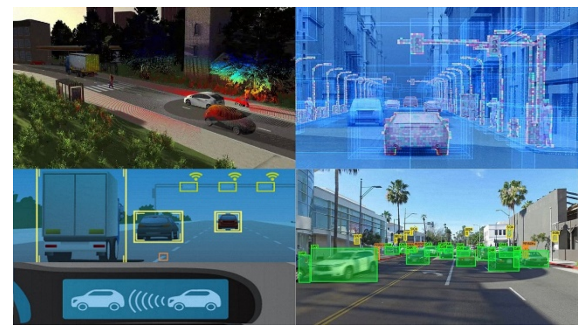


Scene Understanding

- Self-driving vehicles use multiple sensors to perceive driving scenes. Sensor fusion leverages ML and integrates multi-modal sensor data into a precise understanding of the driving env



- ML perceives the real world:
  - Large scale of 2D obj recognition is better than human (what objs are around)
  - 3D scene understanding
    - Depth info is critical for driving
    - 3D model is far more complicated
  - Semantic segmentation: extend of the obstacles
    - Understand the extent of the obj and each pixel of it
    - Eventually 3D world segmentation
  - End-to-end learning: from raw data direct to behavior, like a robot



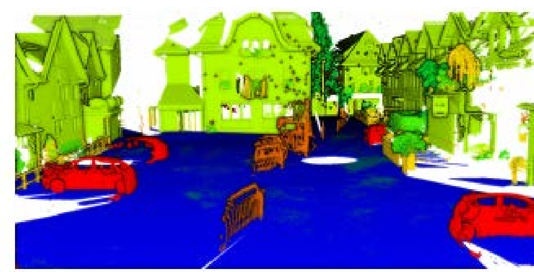
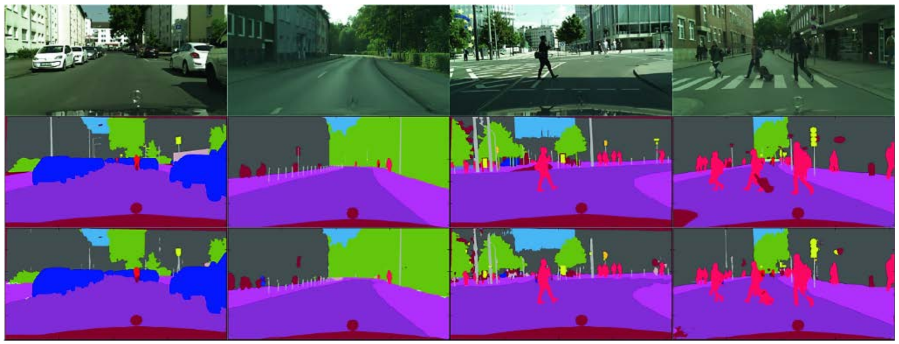
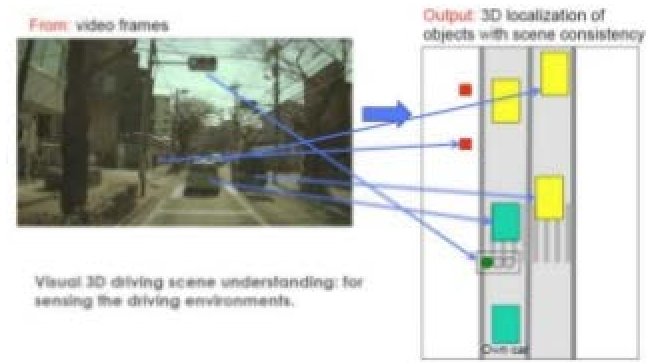


Fig. 3. Example of classified point cloud on Semantic3D test set (blue: man-made terrain, cerulean blue: natural terrain, green: high vegetation, light green: low vegetation, chartreuse green: buildings, yellow: hard scape, orange: scanning artefacts, red: cars).

Scene Understanding: Examples



Visual scene understanding



Building blocks for visual 3D scene understanding towards autonomous driving

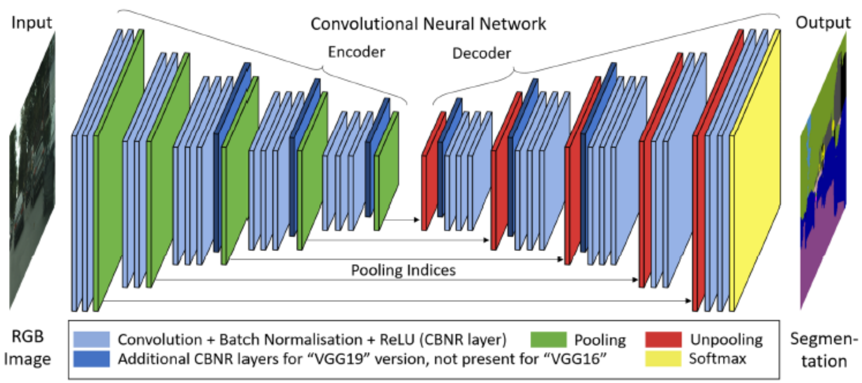
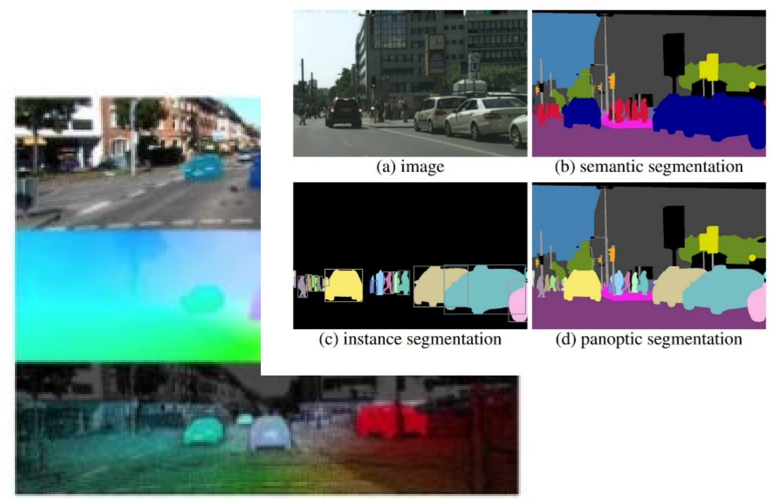


Fig. 1 - Architecture of the SegNet network, with depth variation. Adapted from [1].

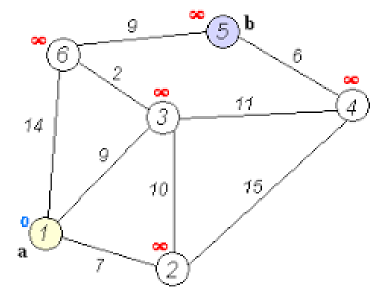
SegNet: A deep convolutional encoder-Decoder Architecture for image segmentation



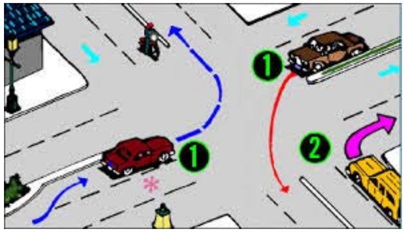
Object scene flow for autonomous vehicles

Motion Planning

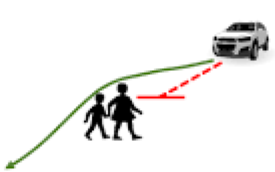
- Mission Planner for long term, focusing on map-level navigation
  - How to navigate from A to B on the map
  - Shortest distance algorithm, time preference, cost preference, etc

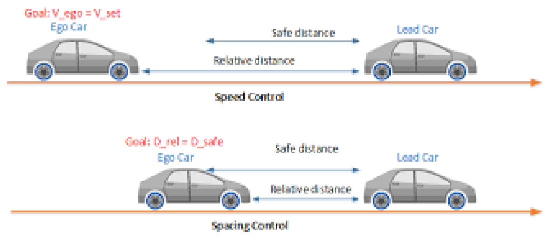


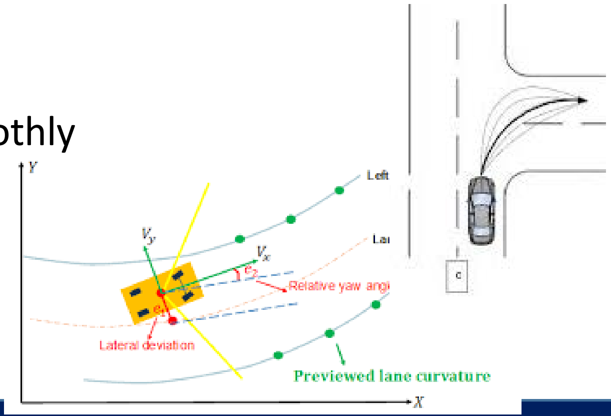
- Behavior Planner for short term, focusing on other agents, rules of the road,, and driving behaviors
  - Driving w/o traffic: keep forward, turn left, turn right, accelerate, decelerate to consider static objects and follow regulatory elements (traffic lights, stop sign, etc)
  - Driving w/ traffic: merge into traffic, change lane, considering pedestrians, vehicles, and cyclists and other moving objects around



- Local Planner, focusing on generating feasible, collision-free paths, and comfortable velocity profiles
  - Immediate or reactive planning
  - A specific path (trajectory) and velocity profile to drive smoothly

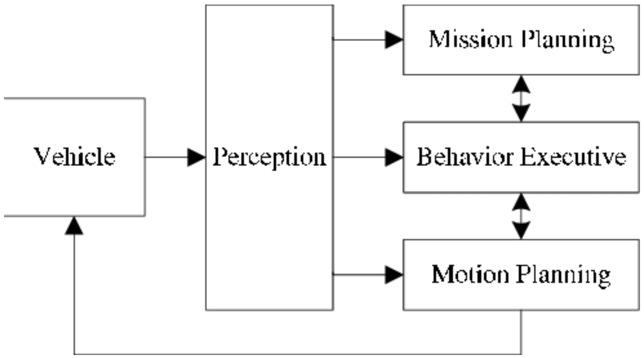






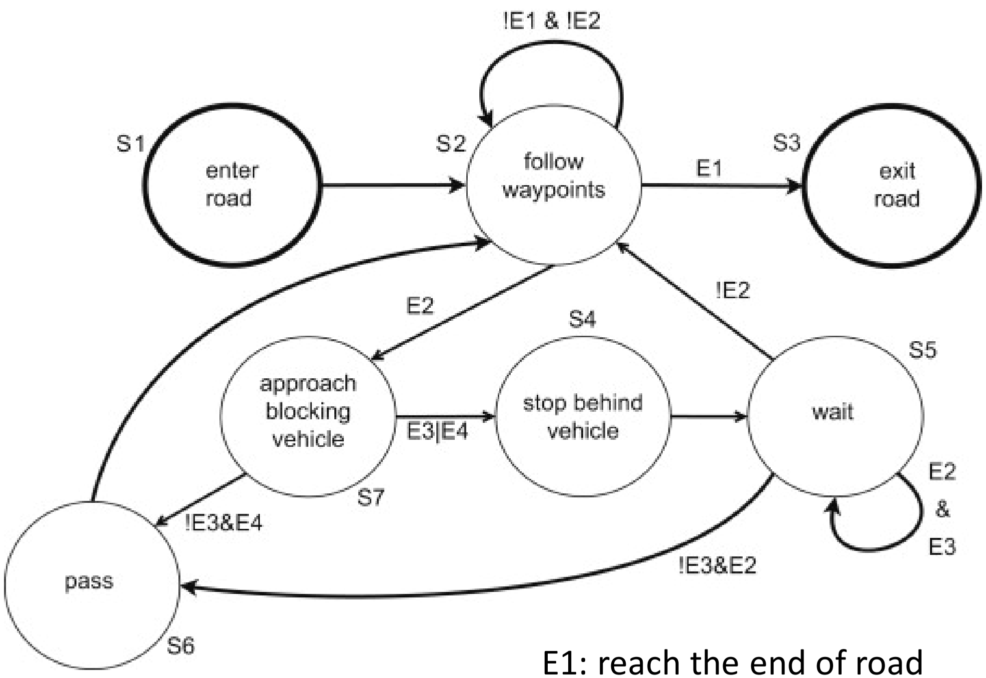
Behavior Planning

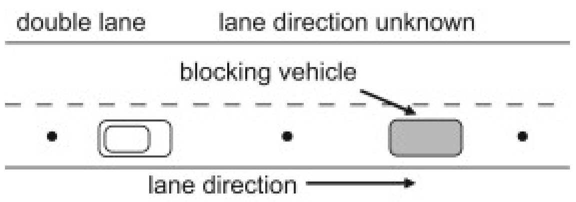
- A behavior planning system plans the set of high level driving actions, or maneuvers to safely achieve the driving mission under various driving situations. Planned path must be safe and efficient
  - Input requirements:
    - HD road map/Occupancy grid, mission path, localization info
    - Perception info:
      - All observed dynamic objs: prediction of future movement, collision points and time to collision
      - All observed static objs: road signs
- Driving maneuvers:
  - Track speed: maintain current speed of the road
  - Follow leader: match the speed of the leading vehicle and maintain a safe distance
  - Decelerate to stop: begin decelerating and stop before a given space
  - Stop at a traffic light: remain stopped in the current position
  - Merge: join or switch one a new drive lane
  - Merge into roundabout
  - Overtake:
  - Change lanes
  - Turn Left/right at intersection
  - Yield
  - Stop-and-go (4-way stop): stop at a stop sign
  - Slow pass
  - Etc



Behavior Planning Methods

- Finite State Machine for BP
  - <CurState, Event, Action, NextState>
  - Each state is a driving maneuver; transitions define movement from one maneuver to another. It must follow certain rules





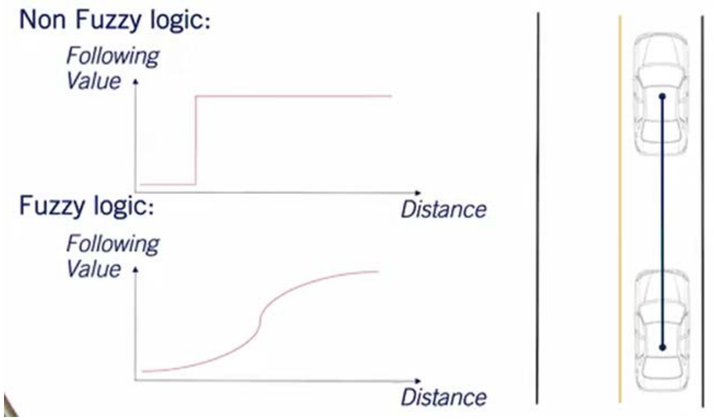
E1: reach the end of road
E2: detect the current lane being blocked
E3: have the passing lane occupied
E4: have a solid lane divider, which means one needs to stp before passing

Advanced Methods for Behavior Planning

Problems with FSM

Rule-explosion when dealing with complex scenarios
Deal with a noising environment
Hyperparameter tuning
Incapable of dealing with unencountered scenarios

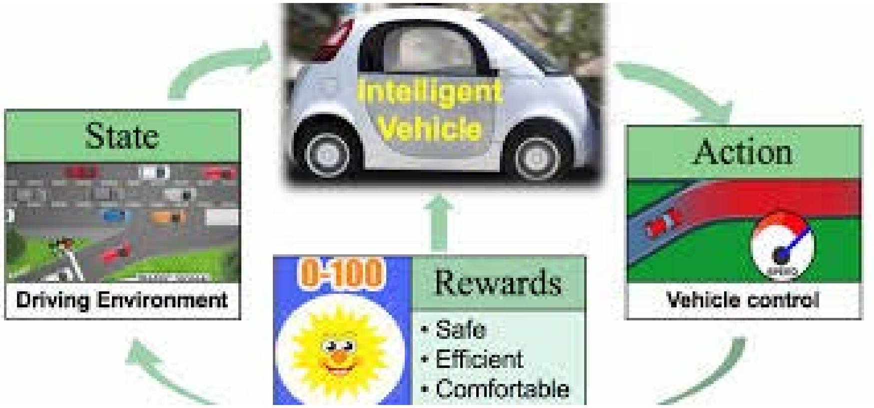
- Rule-based B.P
  - Hierarchy of rules
    - Safety critical
    - Defensive driving to save life, time and money
    - Ride comfort
    - Nominal behaviors
  - Reduced need for duplication
  - Suffer from same challenges in FSM
- Fuzzy Logic
- Reinforcement Learning



Reinforcement Learning for Decision

• RL for Behavior Planning in self-driving

- Agent observes state env
- Agent takes an action to achieve a benefit
- Agent receives a reward based on the results of that action (positive or negative)
- In many cases, the reward may be obtained will in the future
- RL enables the agent to learn the optimal behavior that will maximize the reward



DeepTraffic: Example of RL for Highway Driving

See https://github.com/lexfridman/deeptraffic for online car racing

AD References

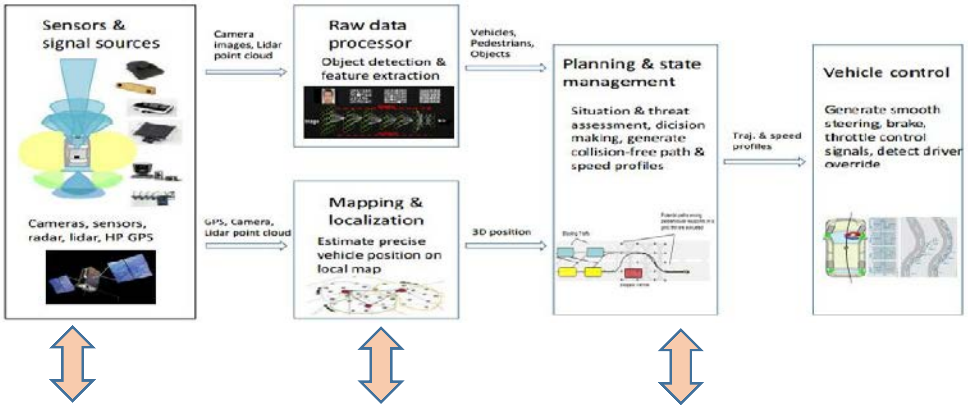
- Levinson (Stanford), et al, Towards Fully Autonomous Driving: Systems and Algorithms, 2011 IEEE Intelligent Vehicles Symp.
- Urmson (CMU), et al, Autonomous Driving in Urban Environments: Boss and the Urban Challenge, J. of Field Robotics, 2008
- C. Reinholtz (U. of Virginia Tech), DARPA Urban Challenge Technical Paper, 2007
- kato, et al, An open approach to autonomous vehicles, IEEE Micro, Nov/Dec 2015
- Bengler (Tech Univ. Munchen), et al, Three decades of Driver Assistance Systems, IEEE Intelligent Transportations Systems Magazine, Winter 2014

Stanley

Boss

Sensing with V2X Communication

- **Vehicle-to-everything** (**V2X**) comm between vehicles, road infrastructure, and pedestrians for road safety, traffic efficiency, and energy savings
  - V2I and V2V
- DSRC: Dedicated Short Range Comm, based on 5.9G 802.11p
  - Toyota in Japan
  - GM in USA in 2016 for V2X
- In 2016, 3GPP launched LTE (cellular V2X) ; 5G enabled V2X
- Cellular-based C-V2X technology in direct communication mode is superior to 802.11p in multiple aspects, such as perf, comm range, and reliability



V2X | edge comp | cloud comp

Holographic Perception → Precise Localization → Intelli Planning Decision → Control and Actuation

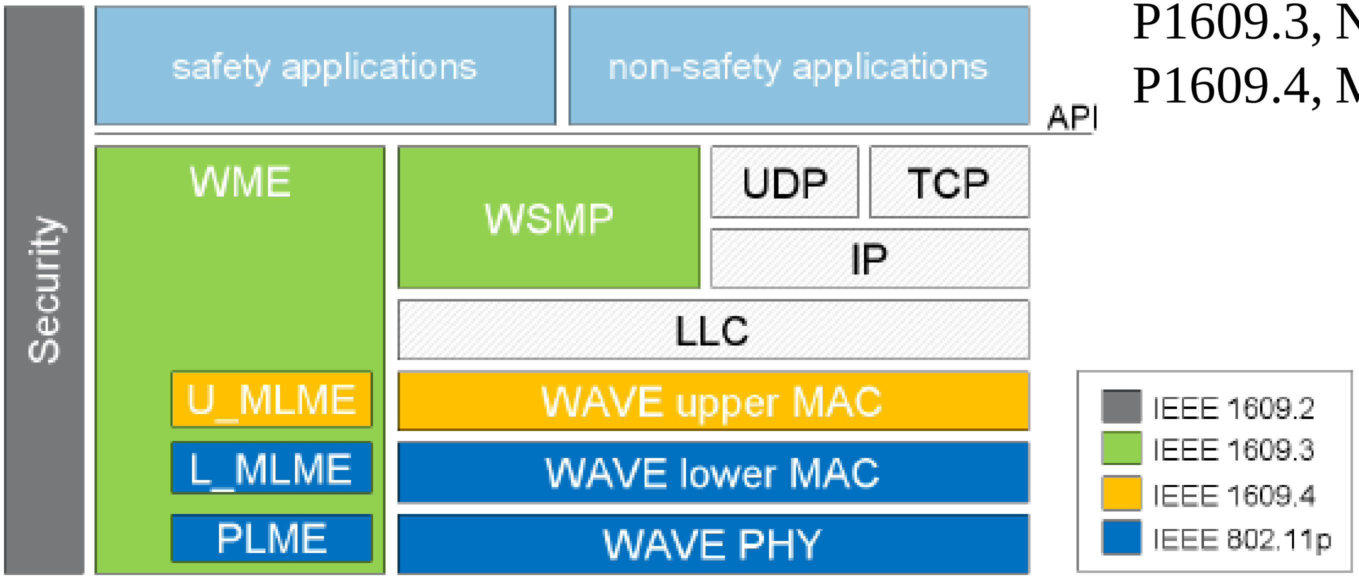
V2X Use Cases: Active Safety

- V2X Comm enables active road safety applications:
  - Forward collision warning
  - Lane change warning/blind spot warning
  - Emergency electric brake light warning
  - Intersection movement assist
  - Emergency vehicle approaching
  - Roadworks warning
  - Platooning: in which vehicles form a convoy behind a lead vehicle driven by a human, could bring a form of supervised automation

WAVE and Communication Stack

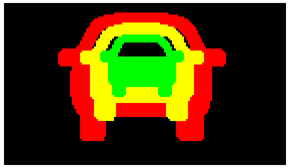
- DSRC: Dedicated Short Range Comm
  - 5.9GHz band allocated by FCC in 1999 and by EU in 2008
  - IEEE 802.11p
- In Dec 2008, IEEE introduced a new standard P1609.5 that defines comm management services.

P1609.1, Resource Manager
P1609.2, Security Services
P1609.3, Network Services
P1609.4, MAC Ext Services



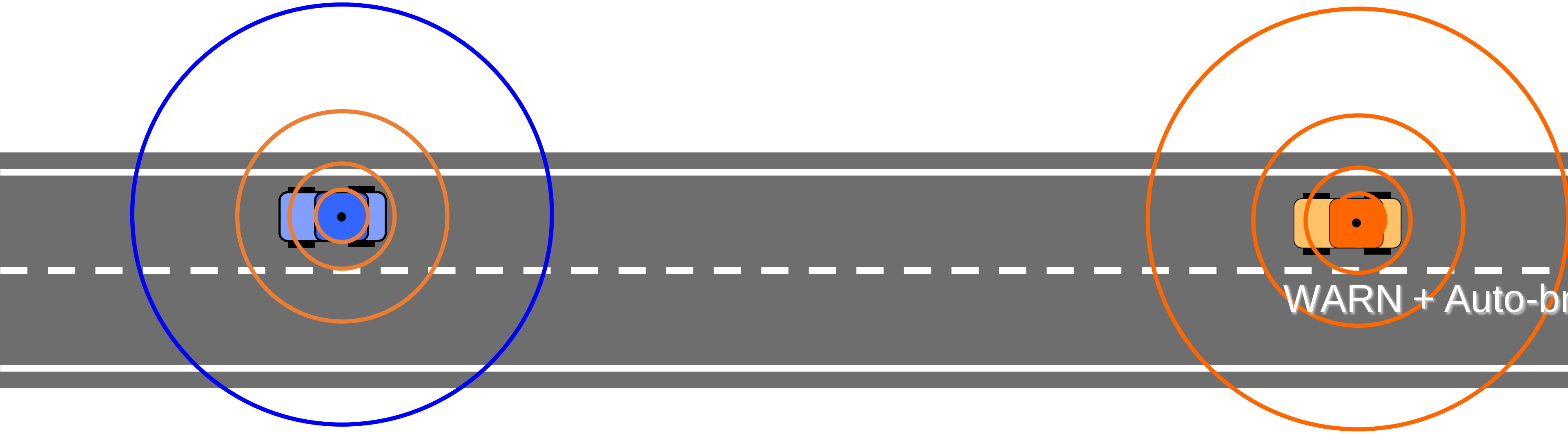
Forward Collision Avoidance w. Auto-Brake

- Host vehicle monitors messages from other vehicles up to 150 m ahead and in the same lane
- Host vehicle warns and then auto-brakes if there is danger of rear-end collision with lead vehicle ahead in lane



+ Haptic Seat

Detected, Caution, Warning

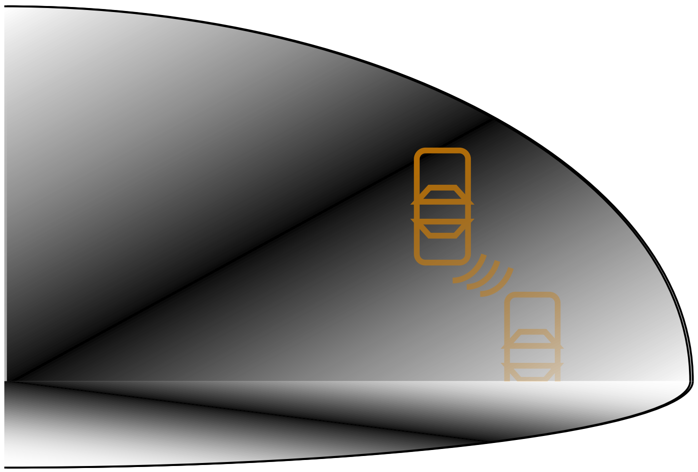


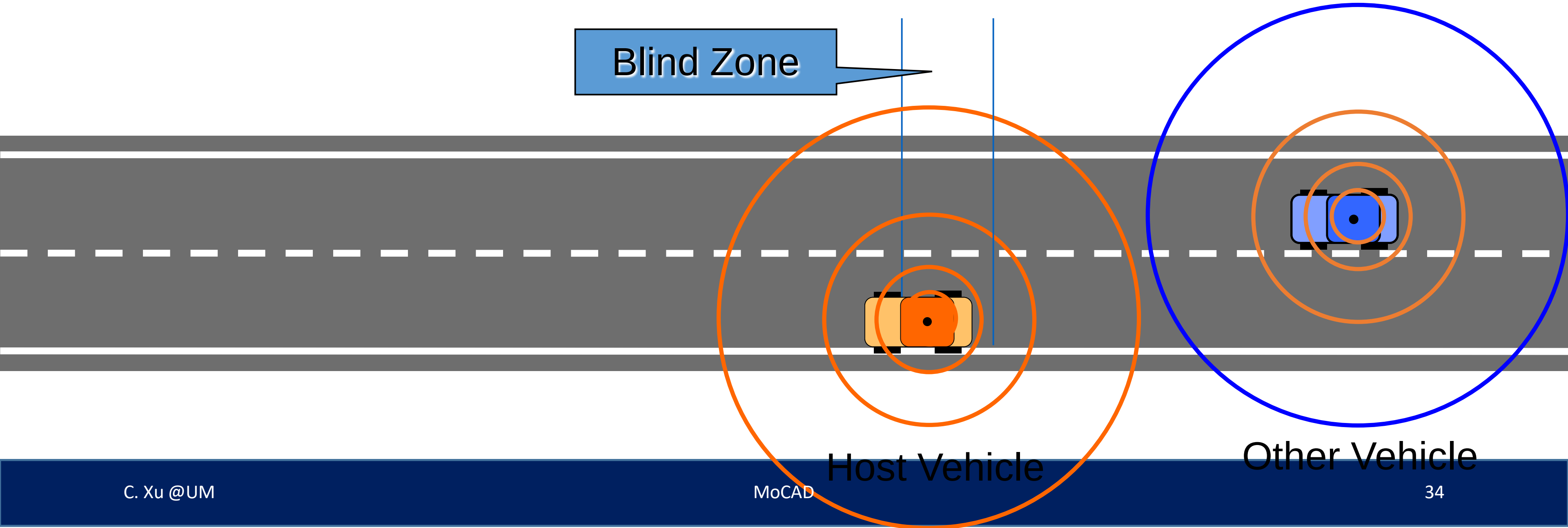
Lead Vehicle

Host Vehicle

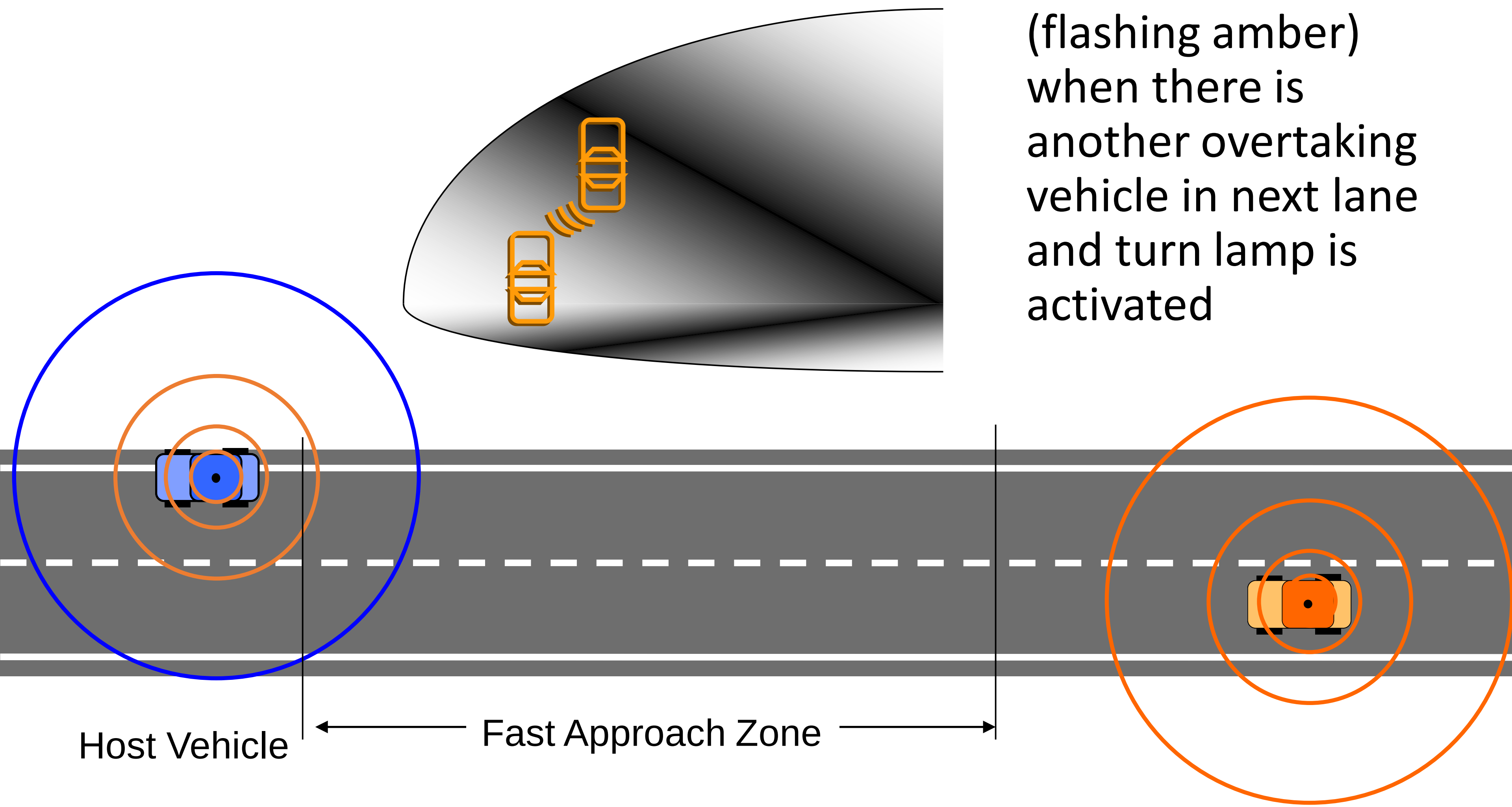
Blind Zone Advisor / Warning

- Host vehicle provides advisory (steady amber) to the driver when there is a vehicle in blind zone
- Host vehicle provides warning (flashing amber + haptic seat) to the driver when there is a vehicle in blind zone and turn lamp is activated





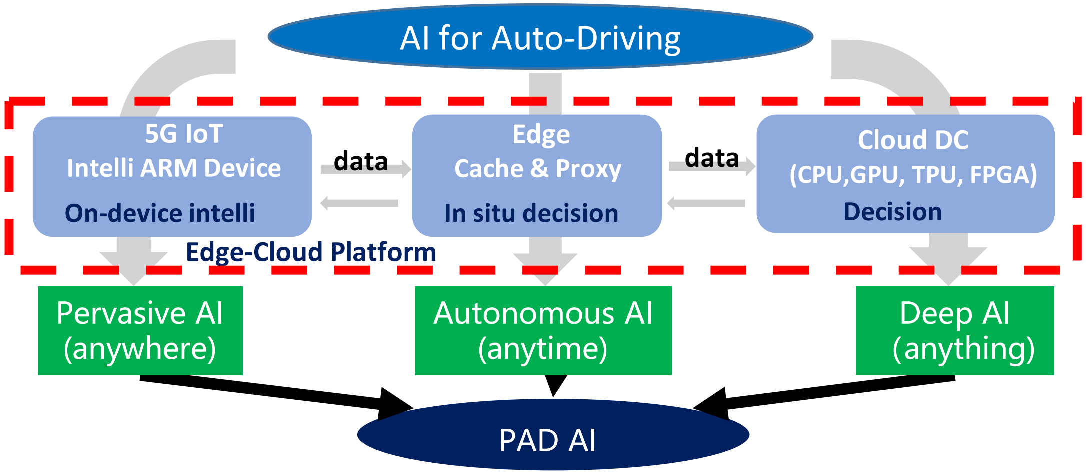
Lane Change Warning - Closing Vehicle



- Host vehicle warns (flashing amber) when there is another overtaking vehicle in next lane and turn lamp is activated

Edge Computing for Connected Vehicles

- Advanced driver-assistant systems (ADAS)
  - Real-time detection of dynamic objects, based on computer vision or deep learning
- Problems and Challenges:
  - Large scale data processing
  - Limited compute resources
  - Unreliable network bandwidth in the moving context
  - Security and privacy
- Cloud-Edge-Onboard Computing Platform with the capabilities of collaboration with other edge nodes, xEdge, and a remote cloud



MoCAD: Macao Connected and Autonomous Driving
澳門協同智能自动駕駛平台及關鍵技術

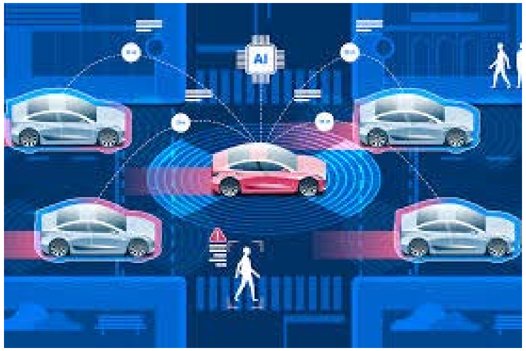
須成忠
澳門大學，智慧城市物聯網國家重點實驗室
2020.6.18

Funded by Macao FDCT 0015/2019/AKP (2020-2023)

澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

使命與願景

- 針對無人駕駛在推廣應用中存在的技術問題，利用人工智能、大數據、智能傳感、車聯網及協同智能等技術手段，研究在**開放不確定環境及大規模複雜混合（自動駕駛與人工駕駛並存）場景**中的無人駕駛核心關鍵技術，打造粵港澳大灣區一流的無人駕駛**群體智能**車載平台和**車路協同無人駕駛**試驗基地



無人駕駛

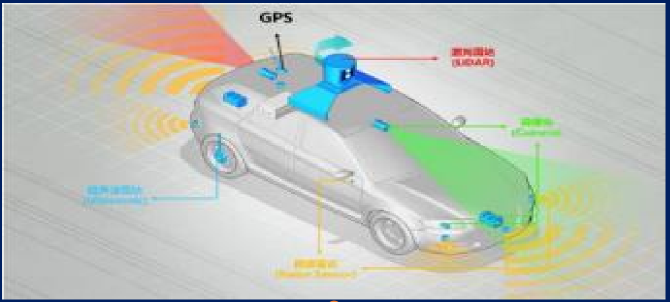
- AI的標志性技術之一，未來汽車工業發展的必然趨勢，是智慧城市的重要一環。

無人駕駛汽車

- 複雜的軟硬件結合的智能系統，運用到了自動控制技術、現代傳感技術、計算機技術、信息與通信技術以及人工智能技術等。

安全價值
避免因駕駛員失誤導致的交通事故

資源價值
通過導航實時監控，減少交通擁堵



環保價值
最優化控制，減少溫室氣體排放

社會價值
服務老年人、殘疾人等弱勢群體

其他價值
促進產業、服務、管理等協同發展

無人駕駛汽車發展歷程

MoCAD

□無人駕駛已成爲世界各國爭奪智能時代技術主導優勢的高地

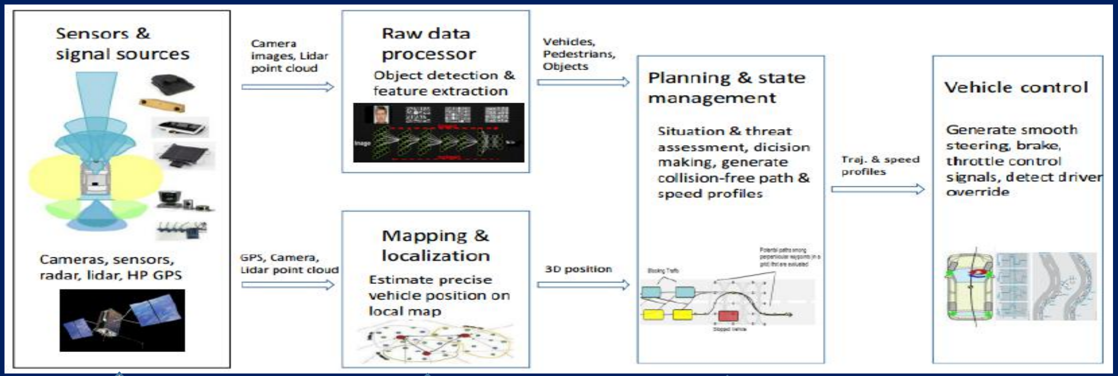
国外无人驾驶汽车发展历程
（DAPPA、Google、Tesla等爲代表）

国内无人驾驶汽车发展历程
（國防科大、百度等爲代表）

無人駕駛關鍵技術

MoCAD

- **單車智能：** 利用車載傳感器感知車輛周圍環境，控制車輛的轉向和速度，根據實時路況進行動態路徑規劃，實現車輛自動、安全的行駛
  - **核心模塊：** 環境全息感知、精准定位導航、智能決策與規劃及高效**控制與執行**。
- **多車智能及車路協同：** 利用V2X及5G網絡通信技術，雲計算技術和邊緣計算等技術，進一步拓展感知範圍和精度，提升決策和規劃的能力



V2X　　edge comp　　cloud comp

環境全息感知 → 精準定位导航 → 智能決策與規劃 → 控制與執行

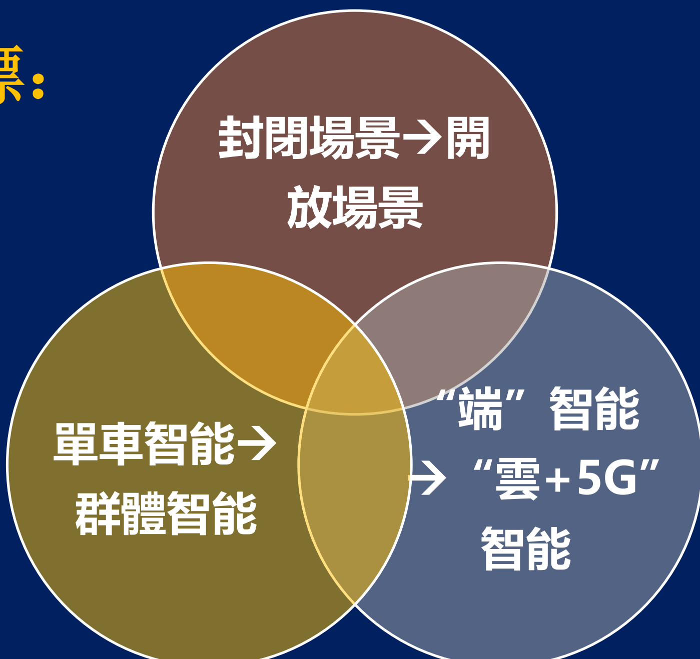
MoCAD目標

MoCAD

❑ 當前無人駕駛產業化和廣泛應用還存在諸多挑戰

- 封閉場景無法窮盡現實世界的各種複雜場景
- 單車智能駕駛技術無法解決大規模車群的協同控制問題
- 尚未形成完整的多車通信鏈路和多車雲協同控制技術

❑ 目標:



科學問題一
大規模混合場景下無人網聯車的群體智能協同控制

科學問題二
開放不確定環境下智能駕駛的高魯棒性及自適應性

澳門大學 UNIVERSIDADE DE MACAU UNIVERSITY OF MACAU

整體技術框架

MoCAD

功能需求分析	開放環境：感知可靠性、控制適應性	大規模場景：協同控制調度、群體智能決策
科學問題	開放不確定性環境下智能駕駛的高魯棒性及自適應性	大規模複雜場景下無人網聯車的群體智能協同控制
基礎研究	智能車聯汽車視覺及信息感知；智能網聯汽車的人機交互機制與方法；智能汽車的人機交互場景數據庫構建	面向多車智能群體協作的組織架構；大規模開放環境下的自學習行爲模式；複雜環境下的智能車聯激勵機制
系統平臺構建	開放動態環境下大規模群體智能技術演示系統；智能網聯汽車視覺感知與人機交互演示系統	基于阿波羅車載操作系統原型無人車一台；基于阿爾法巴無人駕駛巴士平臺原型一套
驗證測試	車雲協作群體智能應用演示；可靠視覺感知和人機交互測試與驗證	車載操作系統實際道路測試；智能網聯車路協同控制演示

合作伙伴

mCAD

□澳门大学，国防科技大学，中科院深圳研究院，百度，海梁科技

- 2019年9月22日國家智能網聯汽車測試示範區正式揭牌，百度、海梁科技、深蘭科技拿到全球首張自動駕駛商用牌照
- **2019.7, 澳门大学与海梁签署合作协议**


澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

HAYLION
海梁科技

澳门大学和深圳市海梁科技有限公司
战略合作框架协议

甲方：澳门大学
地址：中国澳门・氹仔・大学大马路・澳门大学
受权代表人：Xu Chengzhong
职衔：科技学院院长

乙方：深圳市海梁科技有限公司
地址：中国深圳市・福田区・莲花街道・紫荆社区・深南大道 6002 号・华丰大厦 2701

組織結構與分工

MoCAD

科學問題:大規模複雜場景下無人網聯車的群體智能協同控制

開放動態複雜環境下大規模群體智能

- 面向多車智能群體協作的組織架構 — 澳門大學、先進院
- 大規模開放環境下的自學習行爲模式 — 澳門大學、百度科技
- 複雜環境下的智能網聯車的激勵機制 — 澳門大學、先進院

分層次多車智能群體的組織形式
多模態的網聯車協同的運行機制

基于遷移學習的協同學習行爲模式
基于强化學習的實時路徑協同優化技術

基于博弈論和獎懲規則的激勵機制
研究低計算的群體智能算法

科學問題:開放不確定性環境下智能駕駛的高魯棒性及自適應性

智能汽車視覺感知與人機交互

- 智能車聯汽車視覺及信息感知 — 澳門大學、先進院
- 智能網聯汽車的人機交互機制與方法 — 澳門大學、國防科大
- 智能汽車的人機交互場景資料庫構建 — 澳門大學、百度科技

高精度的3D目標檢測和跟踪技術
交通信號標志的魯棒性檢測和識別
複雜環境中自適應的道路邊界檢測

感知推理决策和操控一體化人機交互
基于人機交互的協同在綫感知和預測
人機交互信息融合技術

跨任務樣本生成和主動樣本探測技術
基于生成對抗與統計變分的樣本生成

智能車載操作系統及平臺原型

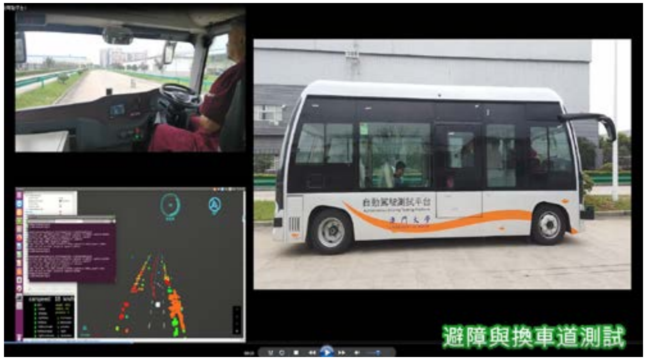
先進院、海梁科技 — 5G驅動的車雲協作機制
- 車雲信息交互與協同技術
- 車雲協作的動態資源管理方法

澳門大學、先進院 — 智能網聯分布式車路協同機制
- 基于認知架構的多車協作技術
- 車路群體協同决策與優化控制技術

澳門大學、百度科技、國防科大、海梁科技 — 面向群體智能的車載操作系統及驗證平臺原型
- 基于群體智能的車載操作系統平臺
- 無人駕駛原型實驗車集成及測試

MoCAD 進展

- 無人駕駛首次落地澳門
  - 政策法規：需要澳門政府允許（1）無人駕駛測試 （2）無人駕駛載人測試
    - 類似法規上海（2019.12），廣州（2019,12），深圳（2020.3.16)
    - 有駕駛員在座監控
  - 澳門交通局法務處發函同意我們載人測試路線和方案（2020.6.8）
- **學校/政府規劃道路： 校園私家路＋校園公共道路**
- **与海梁合作，打造澳门自动驾驶测试平台**
- **与百度合作，打造基于Apollo的澳门无人驾驶小车研发平台**



MoCAD Grand Opening (2020.10.23)

Short Course on

Connected and Autonomous Driving

Objective:

UM is running a research program on Connected and Autonomous Driving, funded by FDCT. CAD Test platforms are to be landed on campus soon. This course is an integral educational component of this program. It’s objectives are to help students

1) Gain basic knowledge and principles about CAD
2) Gain hands-on practical experience in Carla simulation and Baidu Apollo

Format:

- Lectures on principles by knowledgeable professors, plus tutorials by experienced engineers
- Form groups to compete with each other in self-driving car races

Targeted audience and prerequisites:

- Seniors and postgraduates with interest in autonomous driving and connected vehicles
- Good standing of FST/IoTSC/ICI postgraduates; Knowledge in Python is not must, but a plus

Time:

- 19 October to 30 October, 2020 for 10 days
- 9:30am-12:30pm weekday, including 2 hours/day lectures, plus 1 hour/day for hands-on labs

Topical Contents:

1) Overview of CAD (Chengzhong Xu, Chair Professor of CIS)
2) Sensing technologies for self-driving (Hui Kong of NJUST, Guest Lecturer)
3) Visual computing for self-driving (Jiantao Zhou, Assoc. Prof of CIS)
4) Deep ML for autonomous driving (Chengzhong Xu, Chair Professor of CIS)
5) V2X communication and edge computing for connected vehicles (Yuan Wu, Assoc. Prof of CIS)
6) Data and knowledge fusion for CAD (Ryan U, Assoc. Prof of CIS)
7) Modeling and analysis of hybrid human/autonomous driving (Zhenning Li, Research Fellow of IoTSC)
8) Legal, security/privacy and regulation of autonomous driving (Kun Pang Kou, Assoc. Prof of CEE)
9) Tutorial on Baidu Apollo (Baidu engineer)
10) Tutorial on Carla Simulation in Python (TA)

Note that this is a zero-credit hour seminar course, but registration is required because of the limits of class size. Certificate will be issued at the end. Please register the course by scanning the barcode . For inquires, please send email to jtzhou@um.edu.mo

MoCAD Lab Schedule

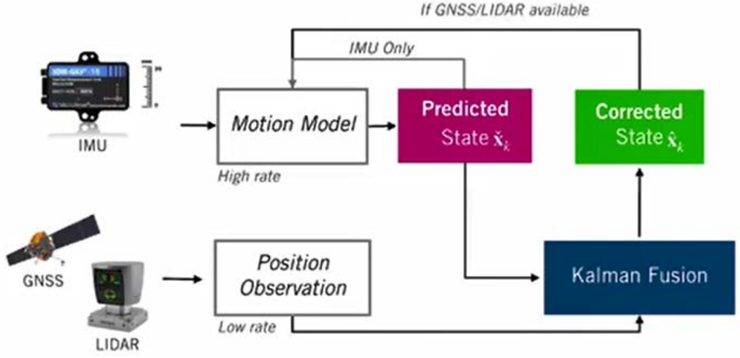
MoCAD Experimental Course Schedule (Carla-python)			
	Course title	Course contents	Projects
D1	Environment setup	1. Course introduction 2. Python Environment anaconda 3. Carla quick start installation and linux build 4. Spawning a vehicle in Carla with your own map (RoadRunner) 5. Carla core concepts	
D2	Running a vehicle by keyboard and collecting data	1. Control a vehicle by apply_control method and keyboard 2. Attach a rgb-image sensor on the vehicle 3. Simulation time-step 4. Try different sensors: RGB-camera, Depth-camera, Lidar, Obstacle ...	**Simple: Sensors** Control a vehicle by keyboard and use Carla python API to collect data from different sensors.
D3	Running a vehicle by PID control	1. Mapping and waypoint 2. Global path planning 3. Local planning 4. PID controller	
D4	Running a vehicle by behavior clone	1. Collecting data 2. Supervised learning 3. Training Neural Network 4. Control a vehicle by the trained NN	**Intermediate: Leader-follower instance** Use the keyboard to control the leader (first vehicle) and the second vehicle follows the leader by PID or behavior clone.
D5	Running a vehicle by reinforcement learning I	1. Introduce the reinforcement learning and DQN 2. Create an Carla environment 3. Building a DQN network 4. Python multi-threading 5. Training the network, agent interacts with Carla environment 6. Control a vehicle by the trained NN	
D6	Running a vehicle by reinforcement learning II	1. Continue action 2. Multi-class regression problem 3. Future work	**Complex: Racing** Use all the knowledge you have learned to control the vehicle so that it can complete a lap on the race road as quickly as possible.

Backup

Sensor Fusion

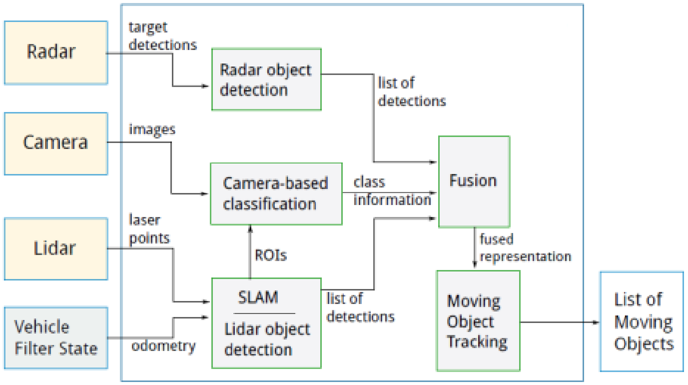
- **Sensor fusion** is combining of sensory data or data derived from disparate sources to reduce uncertainty of sensored info: sensors, soft sensors, history values of sensor data, etc.
  - Data level - to fuse raw data from multiple sources and represent the fusion technique at the lowest level of abstraction.
  - Feature level - features represent information computed on board by each sensing node. These features are then sent to a fusion node to feed the fusion algorithm. This procedure generates smaller information spaces with respect to the data level fusion, and this is better in terms of computational load.
- Calibration: because we're combining info from sensors located in diff places, how to transform all of the measurements so they're expressed in a common reference frame (different sensor model parameters, relative poses and time offsets between sensors)
  - Accuracy: <1 meters for highway lane keep, but GPS accuracy is 1-5m in optimal conditions
  - How fast do we need to update the vehicle state? An update of 15~30 Hz
  - Localization failure: driving in a tunnel w/o tunnel?
- E.g. GNSS with IMU and LiDAR for positioning
  - GNSS provides absolute positioning info to mitigate IMU drift,
  - IMU provides "smoothing" of GNSS, fill-in during outages due tc
  - GNSS tell LiDAR which map to use when localizing

  - Kalman Filter to estimate the state



Fusion for Moving Object Detection and Tracking

- LiDAR + Visual camera
  - Use 3D Lidar sensor generated point-cloud data for clustering (Eulicidean Clustering): aims to obtain the distance to objects rather than to classify them
  - The distance info is used to range and track the objects classified by image processing
  - Sensor fusion needs to calibrate the camera and 3D Lidar. Then, project the 3D point-cloud info onto the image captured by the camera, so that we can add depth info to the image and filter out the region of interest of objet detection
- We can further improve the detection rate using a 3D map and the current position info.
  - Projecting the 3D map onto the image originated on the current position
  - Constrain the region of interest for image processing to this road area so that we can save execution time and reduce false positives.
- Object Tracking
  - Object detection on each frame of the image and point-cloud d
  - We must associate its results with other frames on a time basis trajectories of moving objects
    - Kalman Filter: lightweight and suited for real-time processing
    - Particle Filter: for nonlinear tracking scenarios



Low-level fusion for SLAM, high-level fusion for detection and classification

Bayesian Perception: Basic Idea

- Multi-sensors observations:
  - Lidar, Radar, Stereo camera, IMU, …
- Probabilistic environment model
  - Sensor fusion
  - Occupancy grid integrating uncertainty
  - Probabilistic representation of velocities
  - Prediction models
- Basic Ideas
  - Reasoning at the grid level as far as possible for both:
    - Improving efficiency (highly parallel processing)
    - Avoiding traditional object level processing problems (e.g. wrong detection errors,)
  - Bayesian occupancy filter (BOF)
    - Estimate spatial occupancy for each cell of the grid
    - Grid update is performed in each cell in parallel
    - Extract motion field, using Bayesian filtering and fused sensor data
    - Reason at the grid level (i.e. no object segmentation at this reasoning level)