Droid Equinox 08

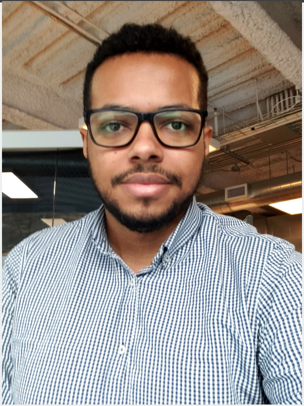
Artificial Intelligence for Autonomous Vehicles

Vinícius Araújo Santos

Machine Learning Engineer **DATA H**

01/23/2020

Received his B.S. degree in Computer Engineering from the Universidade Federal de Goiás in 2015. He received MsC. in Computer Science in Computer Science also at UFG in 2018. Researcher in Machine Learning at DATA H Artificial Intelligence developing solutions for autonomous vehicles . He was a member of robotics team Pequi Mecânico, working in the development of soccer robots between 2013 and 2018.



www.linkedin.com/in/viniciusarasantos

vinicius.santos@datah.ai

+1 905 242-7069

Agenda

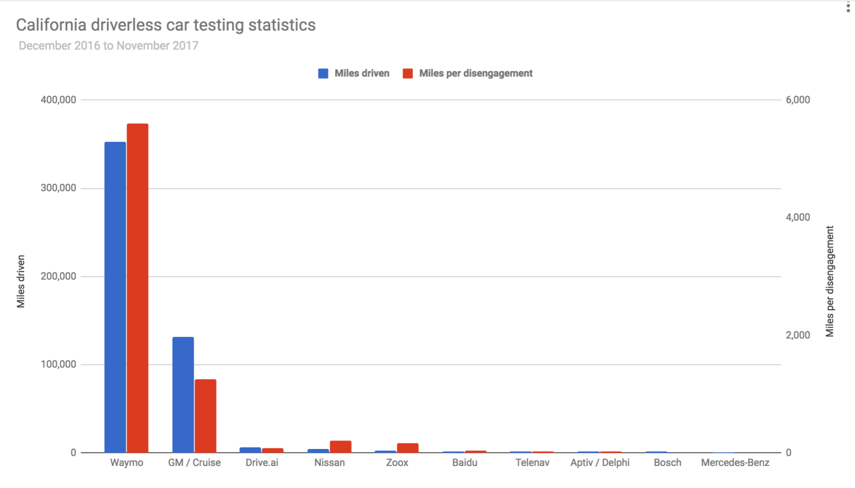
- **Introduction**
  - Why Autonomous Vehicles?
  - Are Emergent Markets ready for this technology ?
  - Is this a reasonable Career to pursue ?
- **The Software Stack of Autonomous Vehicles**
  - Sensing and Scene Perception
  - External Maps and Localization
  - Resources Management and Decision Making
  - Motion and Path Planning
  - Low Level Control

- **The ROS Interface**
  - ROS Basic Structure
  - Topics and Services
  - Using ROS in Python 3
  - Jump Start with ROS Packages
- **Gazebo**
  - Robot simulation
  - Gazebo integration with ROS

Introduction

Why Autonomous Vehicles?

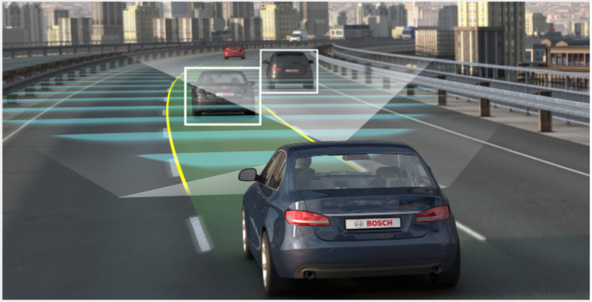
Autonomous vehicles are not the future.
They are here ... and they already work.



The worst **BUG** of vehicles was exposed - and it is us - **the drivers!**

Therefore, to call an Autonomous Vehicle a car is the same as calling the first car a **mechanical horse**.

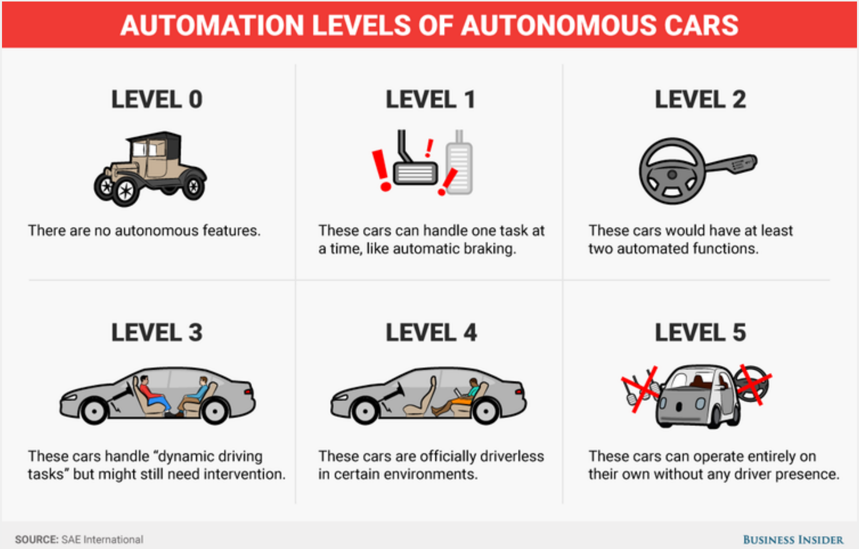
This technology is **SO MUCH MORE** !



Why Autonomous Vehicles?

Traditional cars are this size to carry humans, but small cars can work in **warehouses** and giant cars can work in **mines**.

Why Autonomous Vehicles?



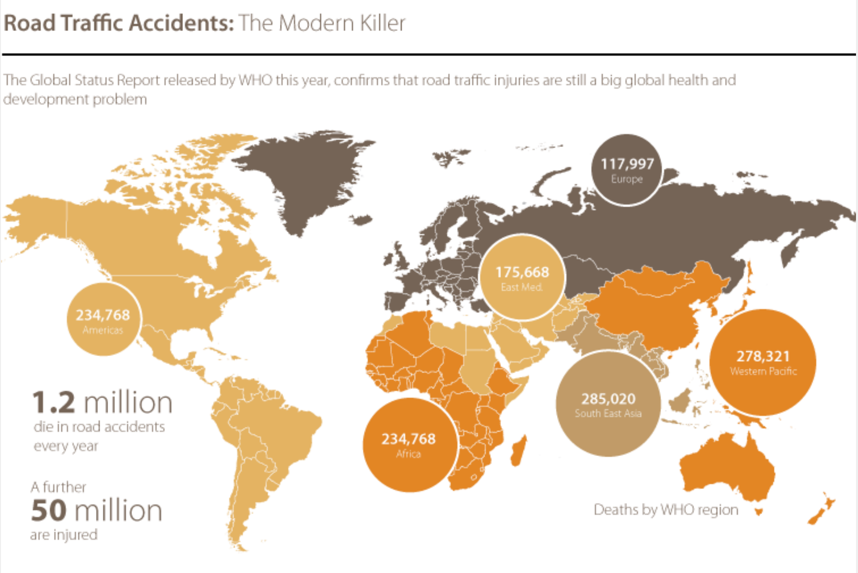
Why Autonomous Vehicles?

How much should you trust a **current** autonomous vehicle?

Why Autonomous Vehicles?

There is this moral dilemma :

But the **REAL** dilemma is this:



But Autonomous Vehicles can save Emerging Markets from Gridlock!

They offer a solution to road congestion **without** the need for government planning or investment (witch may be rare).

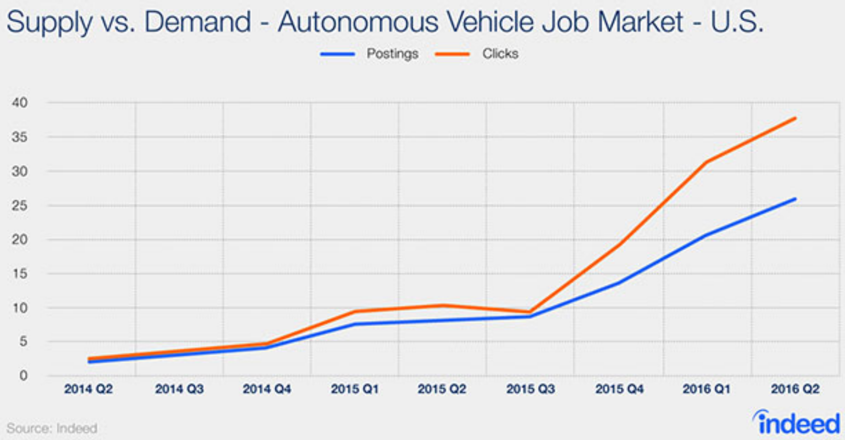
Are Emergent Markets ready for this technology ?

Unfortunately, there are some points to be considered :

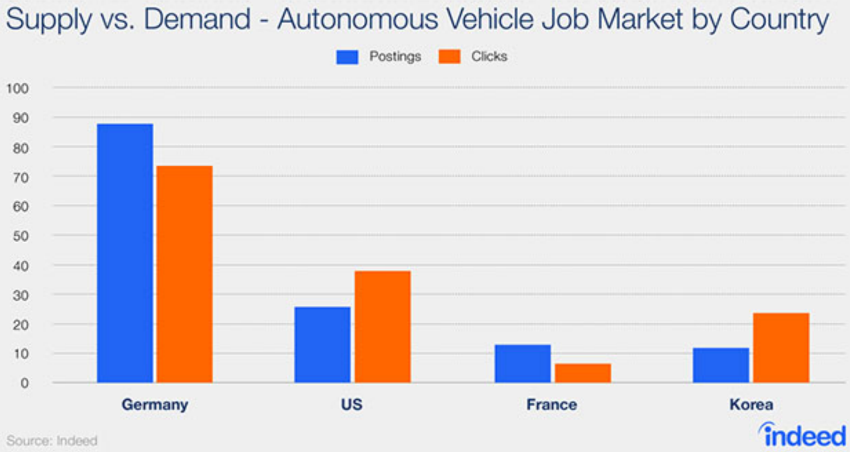
- In most emergent countries, drivers must be **(very)** aggressive to be successful.
- Consequently, will those algorithms be safe in places where human drivers follow different rules-of-the-road?
- If not, how long will a community allow autonomous vehicle-caused casualties before chucking the entire idea?
- The low price-point for transportation in some developing countries may delay the adoption of driver-less services.

Is this a reasonable Career to pursue ?

The self-driving vehicle sector is **attracting**, it's **competitive**, and it's growing **fast**.



There are plenty of places to choose from.



Is this a reasonable Career to pursue ?

If you have access to the internet, you can become proficient at a seemingly endless number of tech-related skills.

If there is something you want to work on, you don’t need to wait to get started!
Dive in, build a portfolio, and demonstrate interest!

The Software Stack

An *AV* relies primarily on three functional blocks:

PERCEPTION

Sensor Fusion, Localization and Scene Understanding.

DECISION MAKING

Path Planning, Context Reaction and Forecast of future interactions .

ACTUATORS

Low Level Control - "Steering, Gas and Brake control"

An *AV* relies primarily on three functional blocks:

PERCEPTION

Sensor Fusion, Localization and Scene Understanding.

DECISION MAKING

Path Planning, Context Reaction and Forecast of future interactions .

ACTUATORS

Low Level Control - "Steering, Gas and Brake control"

Sensing and Scene Perception - Ultrasonic

ULTRASONIC SENSORS*

*HOW DO THEY WORK?

Imitating the navigation process of bats, ultrasonic sensors send out sound waves. When the waves hit an object they produce echoes – revealing the exact location of the obstacle.

WHAT ROLE DO THEY PLAY FOR AUTOMATED DRIVING?

Vehicles use the sensors to detect obstacles in the immediate vicinity – be it cars, pedestrians or bollards. They play an important role for automated parking.

WHAT NEEDS TO BE IMPROVED?

Currently, ultrasonic sensors can only be used at very low speeds. They serve their intended purpose and there is currently no further development needed.

Example - Bosch Ultrasonic Sensor for *AVs*

Sensing and Scene Perception - Ultrasonic

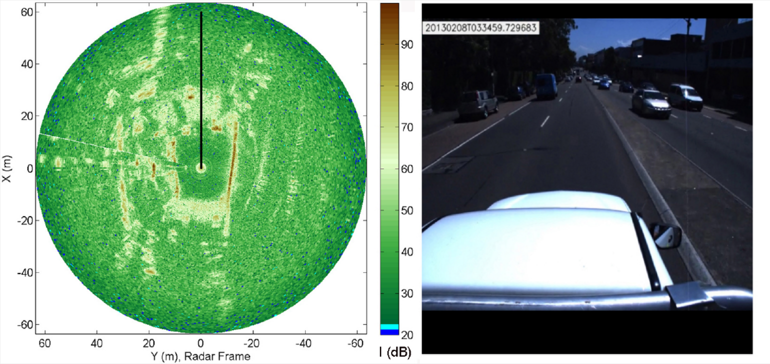
- Based on sounds waves with frequencies higher than that audible to the human ear.
- These sensors are suitable for short to medium range applications at low speed.
- Using echo-times from sound waves that bounce off nearby objects, the sensors can identify how far away the vehicle is from said object.
- Automakers are already using these sensors, albeit only for the short range applications.
- For example, Tesla's Model S sedan is equipped with 12 long-range ultrasonic sensors that provide 360-degree vision to augment the forward facing RADAR, in order to enable its Autopilot system.

Example - Velodyne 3D Lidar for *AVs*

- LiDARs are *"light-based radars"* that send infrared laser pulses and ascertain their return time to create a 3D profile around the vehicle.
- Unlike cameras and radars, LiDARs do not technically detect the nearby objects; rather they "profile" them by illuminating the objects and analyzing the path of the reflected light.
- This, when repeated over a million times per second, yields a high resolution image.

- Since LiDARs use emitted light, its operation is not impaired, notwithstanding the intensity of ambient light which means same intensity in night or day, clouds or sun, shadows or sunlight.
- The result is a greater accuracy of perception and high resilience to interference.
- Currently, the LiDAR sensors come at a very high price. The LiDAR alone makes the entire sensor suite that goes into a vehicle exorbitantly high.
- For example, the Google driverless car features a high-quality Velodyne's LiDAR costing 75,000 USD.
- In future, with solid-state technology coming in, the cost will come down drastically making LiDARs indispensable for any AV.

- Example of radar image (Left). Note that the forward direction of the vehicle is marked by the black line.Front view of the vehicle (Right), as acquired by a co-located camera



Sensing and Scene Perception - Radar

- Both short-range and long-range automotive-grade RADARs are used (mostly in the narrow-band i.e. 27-77 GHz) for AV applications.
- Short-range radars "senses" the environment in the vicinity of a car (30m) and, especially at low speeds; whereas, long-range radars cover relatively long distances (200m) usually at high speeds.
- The radar acquires information from nearby objects like distance, size, and velocity (if it is moving) and can be used to warn the driver if an imminent collision is detected, for example.

Sensing and Scene Perception - Radar

- The high-precision and weather-agnostic capabilities of radars make them a permanent fit for any autonomous vehicle prototype, notwithstanding the ambient conditions.
- Going forward, with the introduction of ultra wide-band radar technology (high frequency 100 GHz), radars will provide more accurate information, be smaller, cheaper and more reliable.

Sensing and Scene Perception - Cameras

Sensing and Scene Perception - Cameras

- Almost all development vehicles today feature some visible light camera for sensory input - many even feature multiple cameras to build a 360-degree view of the vehicle's surrounding.
- Cameras are very good at detecting and recognizing many visual cues in a traffic scene, as these were also created for humans to understand using their visual sense.
- Most approaches take this rich contextual information from images and just feed them into AI-based algorithms for object classification and mild scene context inference. At the same time, some companies, such as Synkar and even Tesla, rely on cameras for almost all of their sensing.

More on Cameras

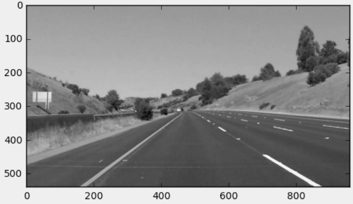
- Cameras are really inexpensive sensor and, even in the wide field, cheap cameras with very high resolution are available – where a LIDAR might see 64 lines, a camera could see 3,000.
- Because of this high resolution, and colour, they are able to understand things about the scene that can't be easily learned from a lower-resolution, grey-scale reading of a LIDAR.
- However, just like human eyes, visible light cameras have limited capabilities in conditions of low visibility.

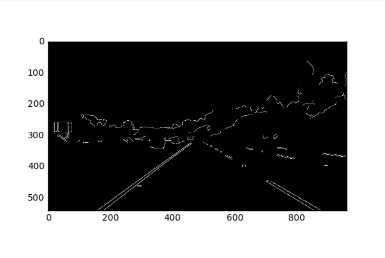
More on Cameras

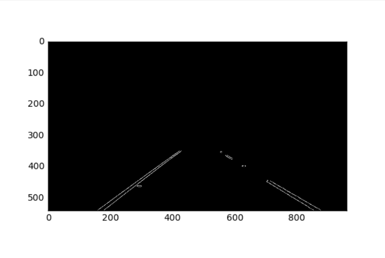
- Additionally, using multiple cameras generates a lot of data to process, which requires substantial computing hardware.
- Much of the excitement in the use of cameras is linked to computer vision and the use of convolutional neural networks (CNN).
- Before the advent of large datasets and general purpose GPUs, classical computer vision were still used in AV applications.

Classic Computer Vision - Lane Detection

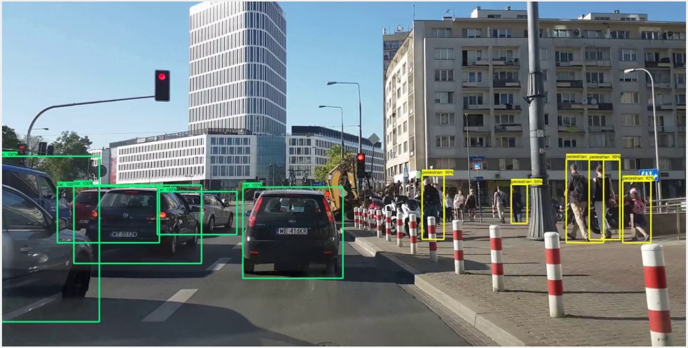
- Canny edge detection is an operator that uses the horizontal and vertical gradients of the pixel values of an image to detect edges. This technique was developed by John F. Canny and still today is a popular edge detection algorithm.







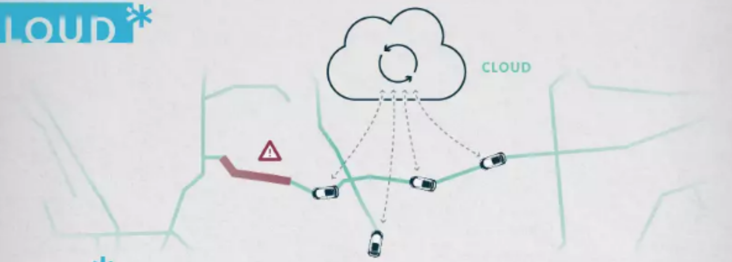
- Recently, CNN based architectures such as the Single-shot detector (SSD) and YOLO have shown great promise in detecting objects more efficiently in a single pass of the image, unlike previous methods.



- Moreover, fully convolutional networks showed outstanding real time performance while maintaining 80%+ mean IoU in datasets such as Cityscapes or KITTI.

Sensing and Scene Perception - Cloud

CLOUD*



*HOW DOES IT WORK?

The cloud serves as a dynamic electronic horizon – it offers highly accurate real-time map data that vehicles draw on. The data is constantly updated by the collective intelligence of the vehicles, for instance reporting closed lanes or defective traffic lights – therefore it also is a type of sensor, providing the vehicle with an image of its surrounding.

WHAT ROLE DO THEY PLAY FOR AUTOMATED DRIVING?

The vehicle's other sensors do not have a reach of more than 250 meters. The cloud data allows vehicles to better anticipate what's ahead. For instance, the vehicle can adjust to an upcoming traffic jam early on, actively reducing consumption and increasing safety.

WHAT NEEDS TO BE IMPROVED?

While prototypes are already using the dynamic electronic horizon, the map data is not yet precise enough for highways and rural areas. Using swarm intelligence of vehicles will only be possible once a sizable amount of connected cars are on the road.

With the recent growth in the autonomous vehicles' market, several companies created HD Mapping suites that leverages Aerial Imagery, Aerial LiDAR data, and Mobile (driven) LiDAR data to create standardized, high-precision 3D base-maps focusing specifically on self-driving vehicle models and markets.

For example, the Sanborn suite offers :

- All road network features are available as precision 3D global data files (GeoJSON, ShapeFiles, and CSV in WGS-84 Ellipsoidal projection).
- All road network features are attributed and classified for the most sophisticated autonomous driving programs.
- Datasets are available for all environments and levels of autonomous drive testing complexity – Freeway, Complex Urban, Complex Parking, etc.

Besides, they also have :

- Complex Urban HD Map datasets to deliver high-precision road feature intelligence:
  - Painted Lines - Painted lines assembled as precision 3D map features, polygons, and lines.
  - Signs - Signs represented as locational precision XYZ features designed to guide the most advanced target recognition for high precision optical localization.
  - 3D Building Models - Buildings constructed and attributed with height points for advanced 3D City Model visualization.
  - Signals and Stop Lines - Traffic signals and stop lines collected in absolute 3D XYZ space delivering precise locations for intersection control protocols.
  - Semantic Data - Lanes, Junctions, Road Segments, DRIVELINES, are assembled with the industry's most robust attribution tables

Localization

Localization is a step implemented in the majority of robots and vehicles to locate with a really small margin of error. If we want to make decisions like overtaking a vehicle or simply defining a route, we need to know what's around us (sensor fusion) and where we are (localization). **Only with this information we can define a trajectory.**

The golden question : *So, How to locate precisely?*

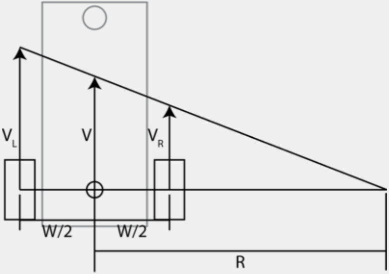
There are many different techniques to help an autonomous vehicle locate itself.

- **Odometry** - Requires an arbitrary starting position and can measure the vehicles’ displacement using cumulative readings of wheel tachometers or/and cameras/computer vision. This technique is generally inaccurate and leads to error accumulation due to measurement inaccuracies, wheel slippage, reading noise, etc.
- **Inertial Measurement Units (IMUs)** - Fusing Accelerometers, Magnetometers and Gyroscopes can also be used to measure the vehicles’s displacement, usually in accelerations, angular velocity and euler angles.
- **Bayesian Filtering** - Kalman and Particle Filters can also be used to fuse different sensor measurements in order to create a robust localization strategy.

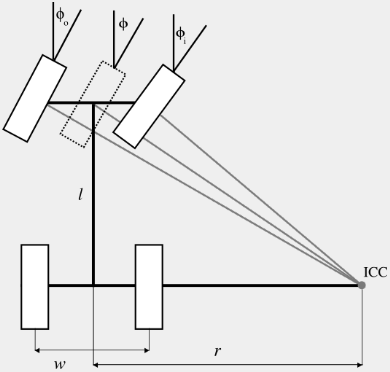
Localization

- **SLAM** - Simultaneous localization and mapping, or SLAM for short, is the process of creating a map using multiple sensory inputs and, at the same time, it also has to figure out where its own self is located in the map. The process of SLAM uses a complex array of computations, algorithms and input filtering to navigate around a previously unknown environment.
- **Probabilistic Localization** - Finally, Probabilistic Localization techniques use a particle filter to track the pose of a robot comparing the current sensory input against a known map.

Usual robotics configurations performs poorly at high speeds and have difficulties in driving straight - as this requires both motors to drive at the exact same speed.

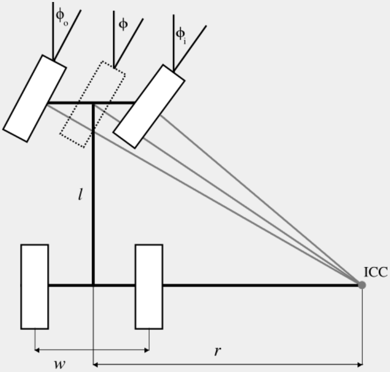


In the other hand, car-like mechanisms are driven by a single motor and can steer their front wheels.



Let the robot coordinate system (x_r, y_r, θ_r) be centered on the car's rear axis.
Then, $\dot{x}_r$ is the car's speed given by the speed of the car's engine. We can now derive expressions for the car's position in world coordinates (x_w, y_w, θ_w).

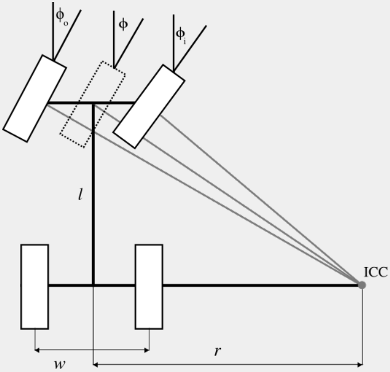
$$\dot{x}_w = \cos\theta_w.\dot{x}_r$$
$$\dot{y}_w = \sin\theta_w.\dot{x}_r$$
$$\dot{\theta}_w = \frac{\tan\phi}{l}\dot{x}_r$$



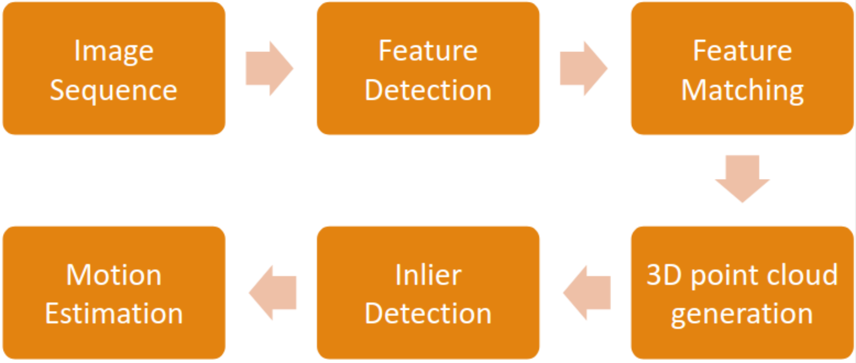
At this point, we can use these equations to predict the car's movement through space as a function of speed and steering angle.

Angles of individual wheels for perfect Ackermann steering can be given as follows:

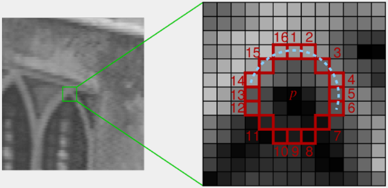
$$\frac{l}{r-w/2} = \tan\left(\pi/2 - \phi_r\right)$$
$$\frac{l}{r+w/2} = \tan\left(\pi/2 - \phi_l\right)$$



The Vision Odometry Pipeline :

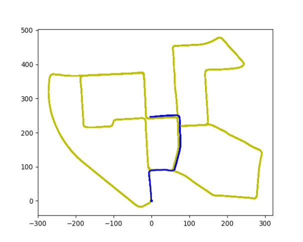


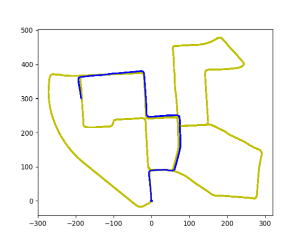
- FAST (Features from Accelerated Segment Test) corner detection
- A Tracker (KLT) is used to find corresponding feature points at T+1
- Triangulation and model optimization to estimate camera movement.

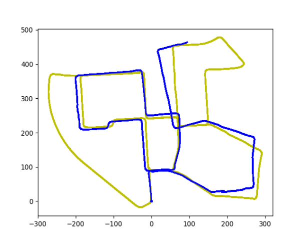


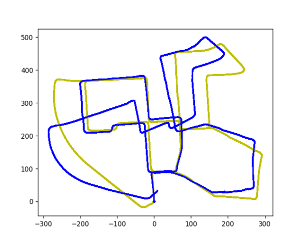
Localization - Visual Odometry

Visual Odometry results through time :

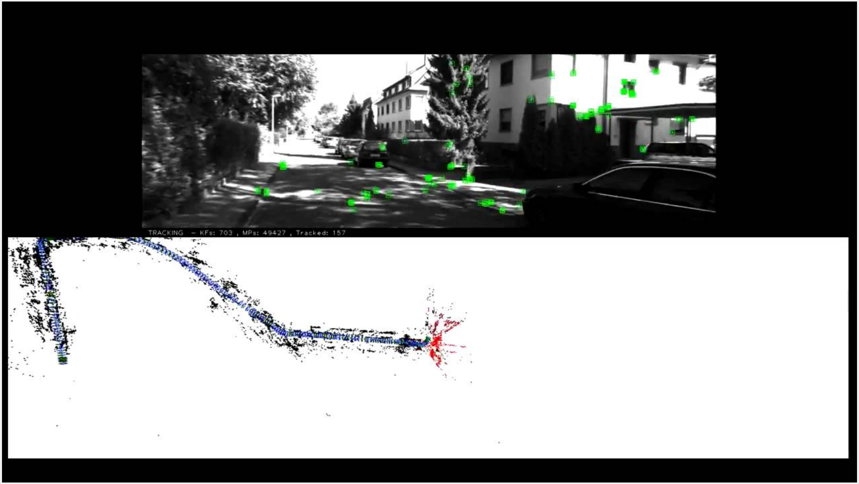








OrbSLAM v2 in the KITTI Dataset :
https://www.youtube.com/watch?v=8DISRmsO2YQ



An *AV* relies primarily on three functional blocks:

PERCEPTION

Sensor Fusion, External maps and Localization for scene perception.

DECISION MAKING

Path Planning, Context Reaction and Forecast of future interactions .

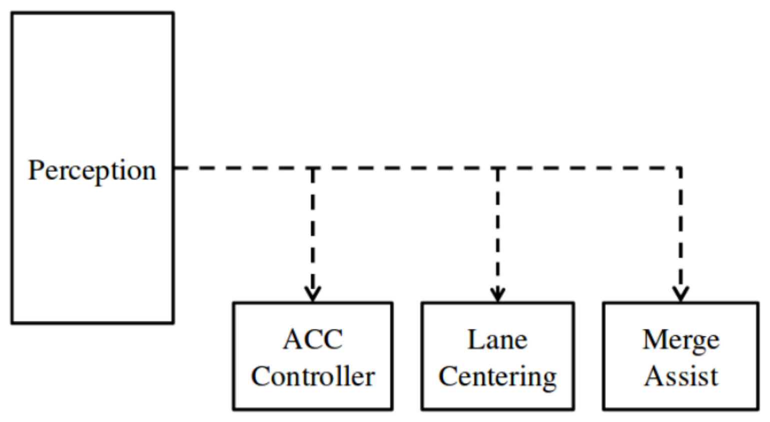
ACTUATORS

Low Level Control - "Steering, Gas and Brake control"

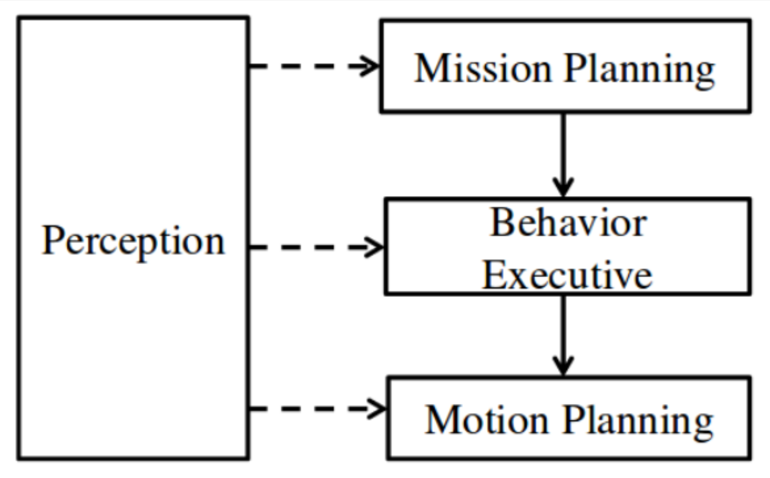
The path planning problem is a well-known NP-hardness where the complexity increases with the degrees of freedom of the vehicle. In most applications, the ground vehicle is reduced to a 2D space of a single plane - where the component is neglected and the vehicle is restricted by some constraints due to the Ackermann configuration. The aim of solving this NP-hardness is

not to find one solution that connects the start point and the goal point, but the optimal solution with the minimum distance and the smoothest maneuvers and without hitting any known obstacles.

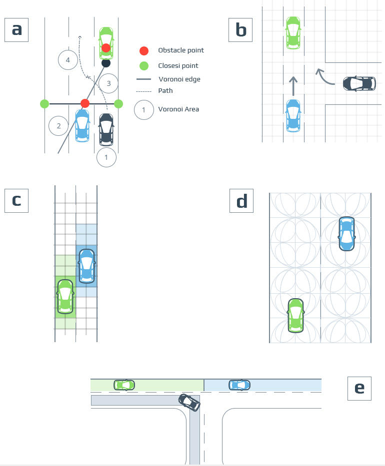
The Parallel path planning framework :



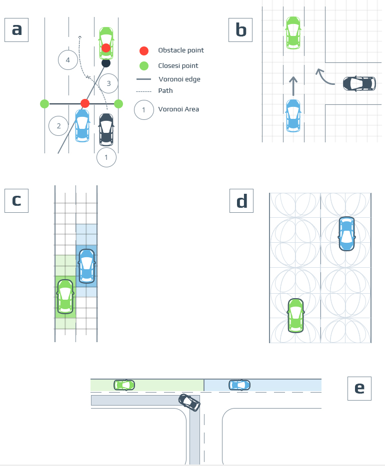
The Hierarchical path planning framework :



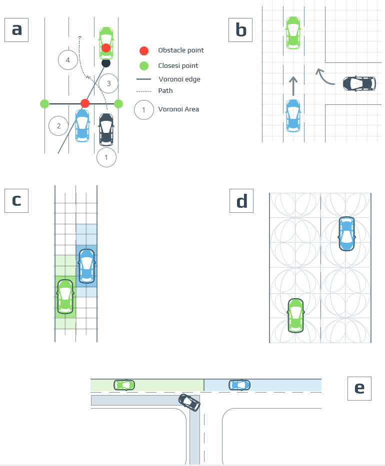
- The Voronoi diagram (a) algorithm generates paths that maximize the distance between a vehicle and surrounding obstacles.
- The occupancy grid (b) algorithm works similarly to the Voronoi diagram, though risk and feasibility are calculated primarily by considering the presence of obstacles and lane and road boundaries.



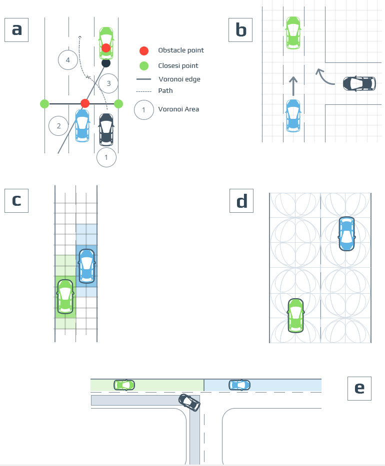
- Whereas the occupancy grid consists almost exclusively of a grid with the obstacle's position, with the cost maps (c) algorithm, the higher cost of a cell results in its more intense representation on the map.



- The state lattices (d) algorithm uses a generalization of grids. Grids are built by the repetition of rectangles or squares to discretize a continuous space, while lattices are constructed by regularly repeating primitive paths that connect possible states for the vehicle.



- The driving corridors (e) algorithm recreates continuous collision-free spaces, bounded by lanes and other obstacles between which the vehicle is expected to drive. Driving corridors algorithms use data from digital maps built by Simultaneous Location and Mapping (SLAM) models.



Usually, the decision making in AV is segmented in two fronts - **Macro** and **Micro** management. This module is often approached

with a **global planner**, which uses a priori information of the environment to create the best possible path, if any, and the **local planner**, which recalculates the initial plan to avoid possible dynamic obstacles. This module has several potential IA

applications and hosts the most secret between developer companies.

While planning for AV, **two** main resources are always expected to be optimized :

- **Time**
- **Battery/Fuel consumption**

These resources are also taken into consideration in the **macro** decision making process (e.g choosing a path for a road trip) in order to provide a better user experience (think AV as the future smartphone).

An *AV* relies primarily on three functional blocks:

PERCEPTION

Sensor Fusion, External maps and Localization for scene perception.

DECISION MAKING

Path Planning, Context Reaction and Forecast of future interactions .

ACTUATORS

Low Level Control - "Steering, Gas and Brake control"

Usually, the control sequence v is obtained by solving a optimization problem. However, the problem is in general **hard** to solve due to safety constraints, as the dynamics of the states x_F from other agents in the scene are **unknown** and the set of safe states R_S is non convex.

When solved, this control sequence is used to **generate** actuators input (eg. gas, brake and steer) using the before-mentioned Ackermann steering model.

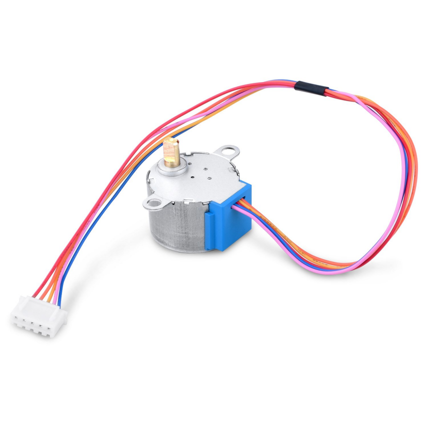
Software Stack Wrap-Up

While the field has made tremendous progress over the last few years, many areas remain inconclusive and several issues recently emerged.

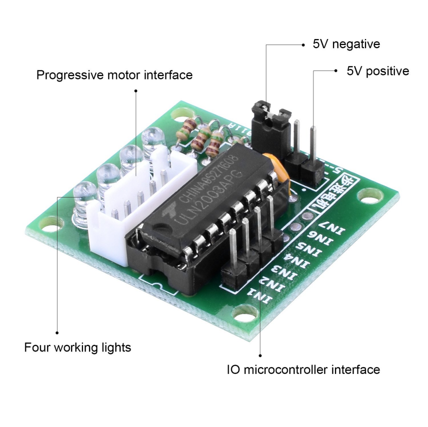
- The increased popularity of data-driven algorithms in both perception systems and planning systems requires a second wave of innovation, requiring even more data and labeling.
- Verifiability, safety, and transparency are key requirements to allow the transition from showcases toward production-ready autonomous vehicles in our everyday lives.
- Autonomous systems still need to reach human-level reliability in decision-making, planning, and perception, and current detection and segmentation accuracies do not yet suffice in difficult conditions, such as intense weather.

Hands-on

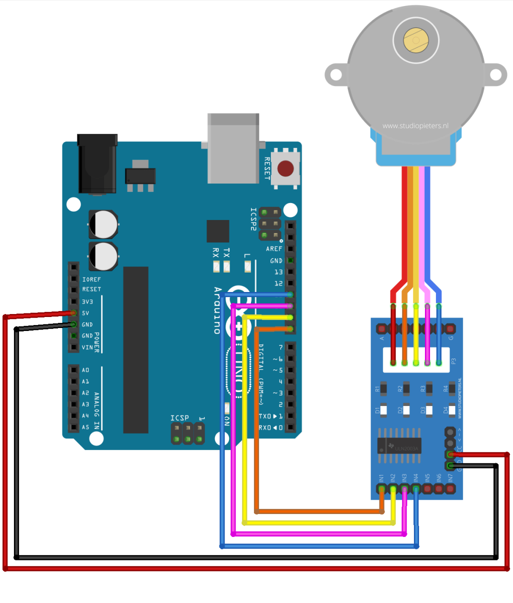
28-BYJ48 Stepper motor



ULN2003 chip driver board



Stepper Motor



```
// Arduino stepper motor control code

#include <Stepper.h> // Include the header file

// change this to the number of steps on your motor
#define STEPS 32

// create an instance of the stepper class using the
Stepper stepper(STEPS, 8, 10, 9, 11);

int val = 0;

void setup() {
  Serial.begin(9600);
  stepper.setSpeed(200);
}
```

```
void loop() {

  if (Serial.available()>0){
    val = Serial.parseInt();
    stepper.step(val);
    Serial.println(val); //for debugging
  }
}
```

Appendix

What is Ubuntu ?

What is Ubuntu ?

- Ubuntu is an open source software operating system that runs from the desktop, to the cloud, to all your internet connected things.
- Complete desktop Linux operating system, freely available with both community and professional support.
- Currently in the 18.04 LTS (Long Term Support)
- If you're considering to pursue a career in Robotics, you should definitely use Ubuntu.

What is Ubuntu ?

- For using ROS and Gazebo we need to use Ubuntu.
- Great opportunity for developing new skills and for our future in I2A2
- **Tutorial:** https://hackernoon.com/installing-ubuntu-18-04-along-with-windows-10-dual-boot-installation-for-deep-learning-f4cd91b58555

Thank You!

www.linkedin.com/in/viniciusarasantos

vinicius.santos@datah.ai

+1 (905) 242-7069