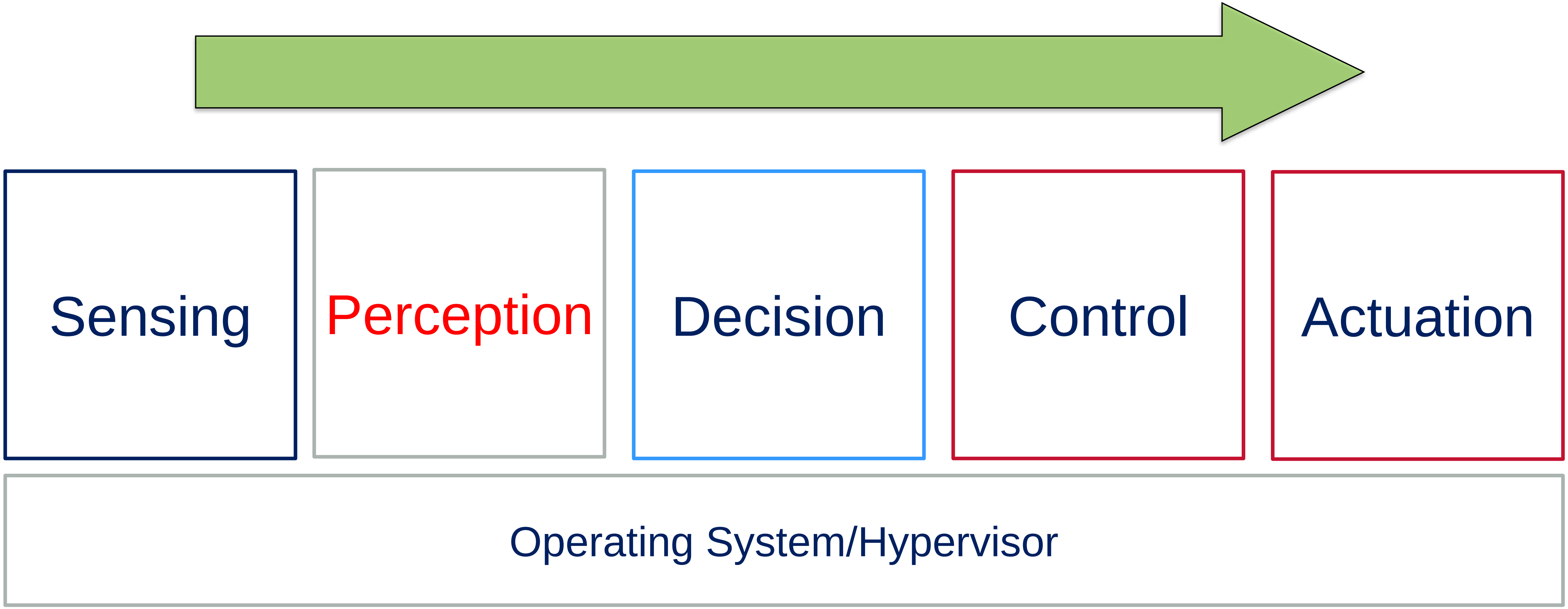
MECH5170M
Connected and Autonomous Vehicles Systems

Perception and Segmentation

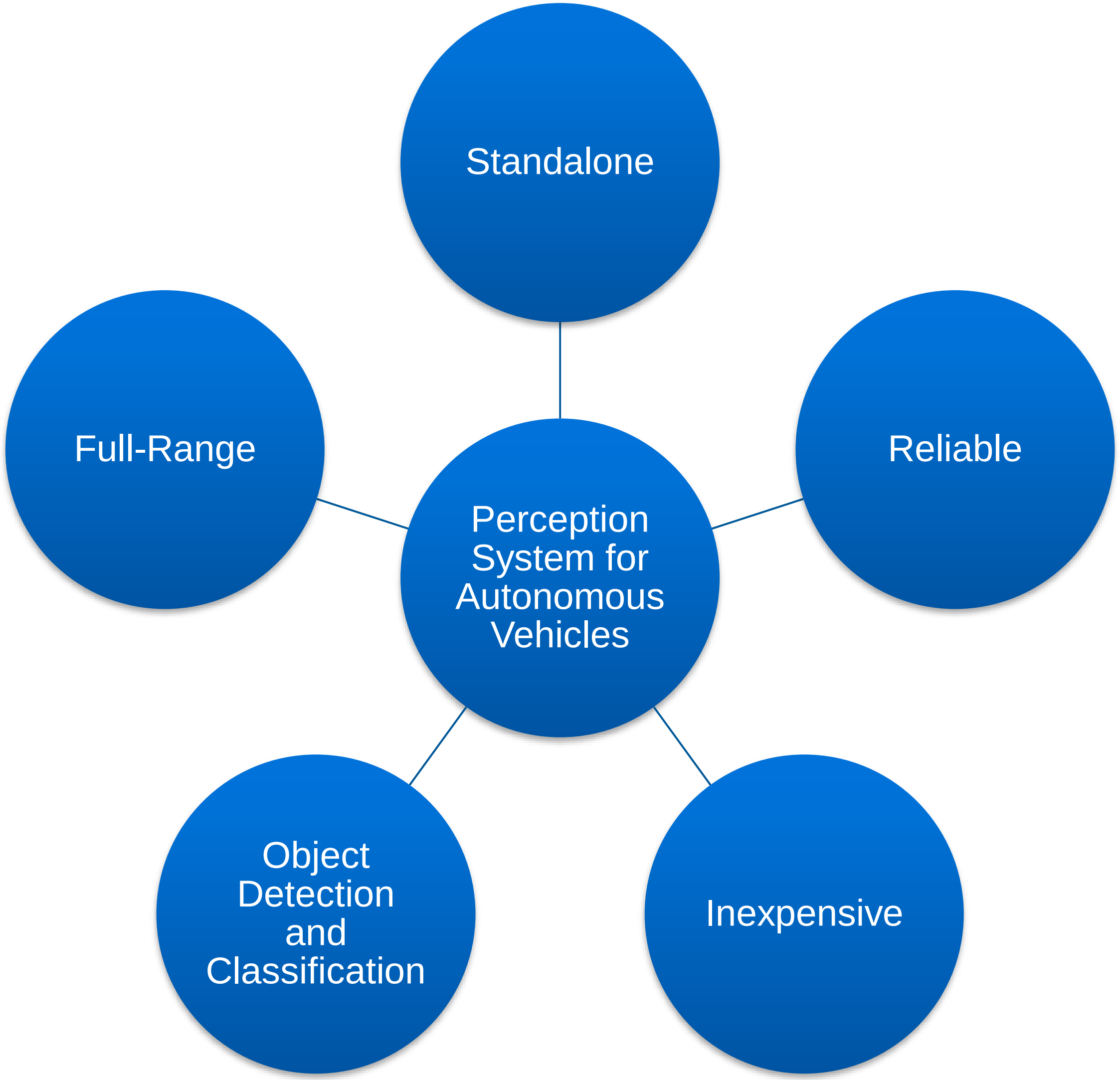
Kris Kubiak (k.kubiak@leeds.ac.uk)

Perception

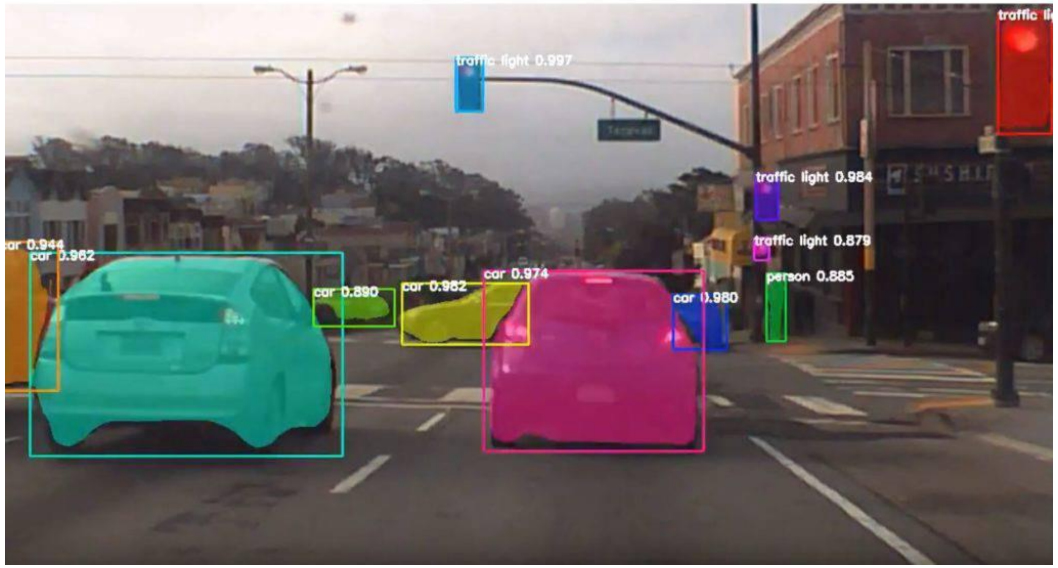
Self-driving car architecture



Perception system requirements



Perception: How Self-Driving Cars 'See' the World



Performance Requirements

Detect objects up to 150 m

Unify sensor data up to 50 m

Acquire sensor data at up to 20 Hz

Detect object size with an accuracy of up to 90%

Detect object distance with an accuracy of up to 95%

Detect object velocity with an accuracy of up to 90%

Classify objects with an accuracy of up to 80%

Estimate vehicle motion with an accuracy of up to 90%

3D Object Detection

Perception

- Localisation
- Object Detection
- Object Tracking
- Traffic recognition
- Road topology identification

Many things fall under the vague category of perception, list to the left is not complete

Localisation:

- Strongly connected to sensor fusion
- May use algorithms such as *particle filters* in addition to Kalman filter
- Could further sub-divide into road-level localisation in a map, or lane-level localisation on a road, or localising within a lane

Perception

UNIVERSITY OF LEEDS

- Localisation
- Object Detection
- Object Tracking
- Traffic recognition
- Road topology identification

Object detection

- Use vision or deep learning algorithms to detect various kinds of objects
- Pedestrians, Bicyclists, other vehicles, traffic signs, lane markings, obstacles
- Objects could be static or dynamic, and detection algorithms may vary accordingly

Object tracking

- Tracking trajectories of moving objects
- Could be based on deep learning or algorithms like optical flow

Perception

UNIVERSITY OF LEEDS

- Localisation
- Object Detection
- Object Tracking
- Traffic recognition
- Road topology identification

- **Prediction**: Generate most probable trajectories of vehicles, pedestrians, obstacles in the environment
- **Reference planning**: Generate trajectories for vehicle + Behavioural planning (traffic-aware): modify according to environment models
- **Obstacle avoidance**: Stopping or manoeuvring around an obstacle

AI v.s. ML v.s. DM

What is the differences between Artificial Intelligence, Machine Learning and Data Mining?

- AI: intelligent agents, agents, agents…
  - Is a system that perceives its environment and takes actions that maximize its chances of success.
- ML: is a part of AI. From empirical data, learn patterns or predictions thought to be features of the underlying mechanism that generated data.
- DM: the analysis step of the “Knowledge Discovery in Databases” process – KDD. Extract information from a data set and transform it into an understandable structure for future use.

Object Detection

Self-driving car architecture

Data Acquisition

Data Representation

Segmentation

Object recognition (AI)

Objects Annotations

Object detection

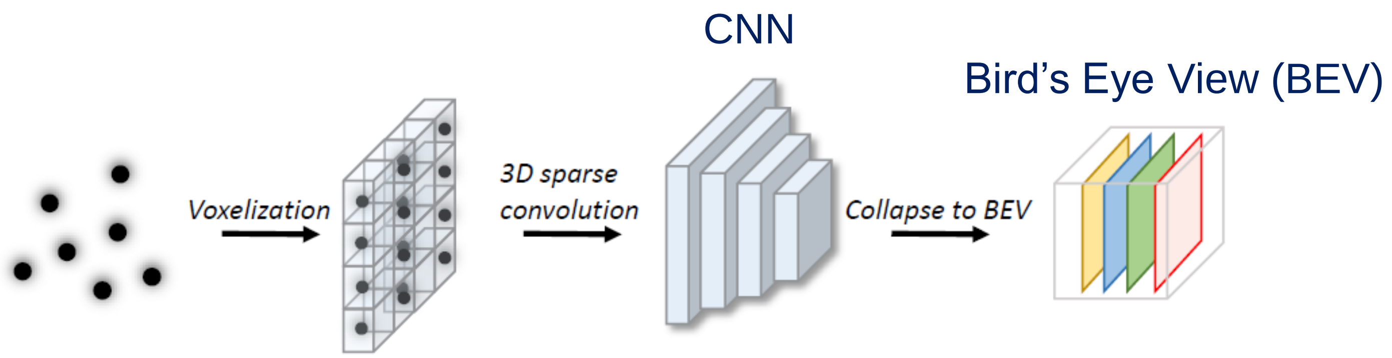
3D object detection methods

Image based Methods
- Result-lifting based
- Feature-lifting based

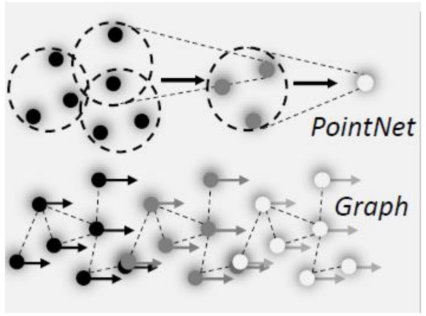
Point Cloud based Methods
- Voxel based
- Point based
- Point-voxel based

Fusion based Methods
- Sequential Fusion
- Parallel Fusion
  - Early Fusion
  - Deep Fusion
  - Late Fusion

Focusing on depth estimation



Fusion often is used to detect failed cases



[Qian et. al. 2022 - 3D Object Detection for Autonomous Driving: A Survey]

Data representation

Following representations for LIDAR data are most popular:

- Point-cloud representation in the 3D space
- Feature representation
- Representation using grids

The choice of representation guides the choice of the algorithms chosen downstream for segmentation/detection

Point-cloud based approaches may need filtering algorithms to reduce number of points

- Voxel-grid filtering : cover the space with tiny boxes, and replace each box with the centroid of the box

Data representation

Feature-based approaches

- Extract specific features from the point cloud such as lines or surfaces
- Most memory-efficient approach, but accuracy subject to nature of the point cloud

Grid-based approaches

- Discretise space into small grids and represent the point cloud as a spatial data structure

Segmentation algorithms

Segmentation: Clustering points into multiple homogenous groups

Broadly divided into:

- Edge-based methods: Good when objects have strong artificial edge features (e.g. road curbs)
- Region-based methods: Based on region-growing, i.e. pick seed points, and then grow regions based on criteria such as Euclidean distance between points, surface normals etc.
- Model-based methods: Fit points into pre-defined categories such as planes, spheres, cones etc.
- Attribute-based methods: first compute attributes for each point, and then cluster based on attributes

Segmentation

UNIVERSITY OF LEEDS

Some popular segmentation algorithms

- **RANSAC (random sample and consensus)**
- **Hough Transform**
- Conditional Random Fields, Markov Random Fields (also used for sensor fusion between LIDAR and vision)

Random Sample and Consensus

Algorithm for robust fitting of a model in the presence of outliers

Given a fitting problem with parameters $\boldsymbol{\theta}$, estimate optimal values for $\boldsymbol{\theta}$

What is a “model”?

- Line, bounding box, etc., i.e. any parametric shape

Assumptions:

- Parameters can be estimated from n points
- There is a total of $m \gg n$ points

RANSAC continued

1. Select n points at random
2. Estimate $\boldsymbol{\theta}$ values for the shape fitted to the above n points (say the values is $\boldsymbol{\theta}^*$, and the resultant shape is now $S(\boldsymbol{\theta}^*)$
3. Find how many of the m points are within some ϵ tolerance of $S(\boldsymbol{\theta}^*)$. Say this is k
4. If k is large enough, accept model and exit with success
5. Repeat 1 to 4 some ℓ times
6. Fail if you get here

Problematic parameters selection: how to choose $\boldsymbol{k, \ell, n}$

Choosing parameters for RANSAC

Pick n based on how many points is required to find a good fit for the shape

Pick k based on intuitively how many points would lie in a shape

- If there are multiple “models” or “structures” with an image, remove the points associated with a shape once RANSAC terminates with success, and then redo RANSAC

Probability that a selected point is an *inlier:* p_g

Probability that an iteration of RANSAC fails without finding a good fit: p_{fail}

Pick $\ell = \frac{\log(p_{fail})}{\log(1-p_g^n)}$

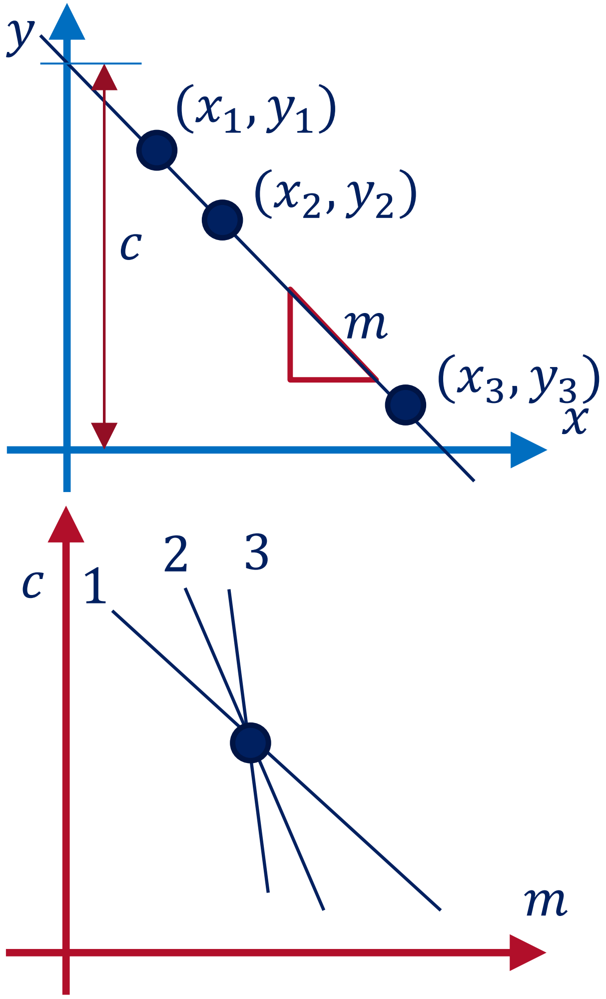
Hough Transform

A tool to detect **lines, circles** and more **general shapes**

One of the tools used for **lane marking detection** from (pre-processed) images

Operates on sets of points and helps obtain a geometric representation of shapes that points may form

We will see how Hough transform works in 2D



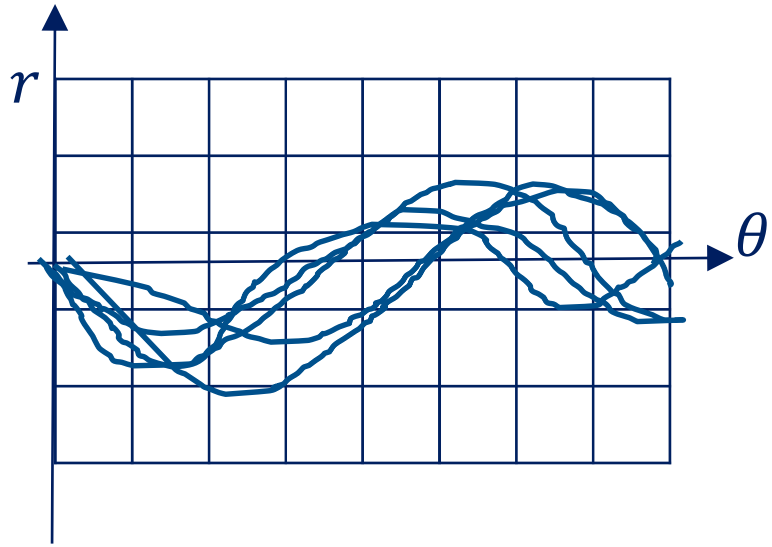
Let's look at a simple transformation:
map (x, y) space to the (m, c) space

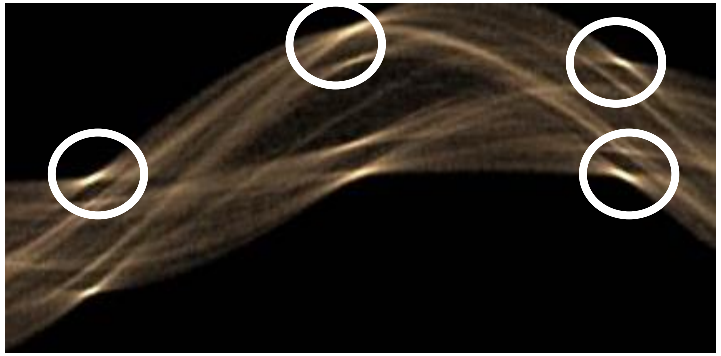
$$\left.\begin{aligned} y_1 &= m_1 x_1 + c_1 \\ y_2 &= m_2 x_2 + c_2 \end{aligned}\right\} (x_1, y_1), (x_2, y_2) \text{ are known!}$$

Plot lines in the (m, c) space: each point maps to a line in the (m, c) space

If lines corresponding to different points intersect: this represents a collection of collinear points with the slope and intercept defined by the intersection

HT in polar coordinates





Problem with the (m, c) space is that $m \to \infty$ for vertical lines. So all lines in the (m, c) space corresponding to points on a vertical line would only intersect at infinity.

Resolved by instead considering the (r, θ) space

Line in (r, θ)-space: $p = x_1 \cos\theta + y_1 \sin\theta$

Here, p = length of normal to line, θ angle made by normal with x-axis

Now point in (x, y) space maps to a sinusoid in (r, θ)-space

To find lines, we let the sinusoids vote: i.e. identify the points in a suitable grid that accumulate weight

HT discussion

Pros:

Conceptually simple and easy to implement

Robust against noise

Handles missing and occluded data very well

Can be adapted to various shapes beyond lines

Cons:

Computationally complex if there are many shapes to look for

Can be fooled by apparent lines

Collinear line segments may be hard to separate

Conclusions

Perception - creation of virtual representation of the world around the vehicle

Main steps:

- Data Acquisition (Lidar, Camera)
- Data Representation
- Segmentation
- Simple object recognition

ANY QUESTIONS

???