MOBILE ROBOTS 2+3 – Perception, Vision and Machine Learning

Prof. Francesco Mondada

EPFL

2025-2026

BIBLIOGRAPHY AND SOURCES

Introduction to Autonomous Mobile Robots (ch. 4) R. Siegwart, and I. Nourbakhsh, MIT Press, 2011

Elements of Robotics (ch.12), M. Ben-Ari, F. Mondada, Springer, 2017.

Mobile Robots - EPFL - J.-C. Zufferey, Felix Schill, 2013

Springer Handbook of Robotics (ch. 32-34), B. Siciliano, and O. Khatib (Eds.), 2nd edition, Springer, 2016.

Elements of Robotics (ch.13-14), M. Ben-Ari, F. Mondada, Springer, 2017.

Reinforcement learning, Sutton & Barto, MIT Press,1998.

Computer Vision

« Computer vision is the **science and technology of machines that see**, where see in this case means that the machine is able to extract information from an image that is necessary to solve some task. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. » Wikipedia

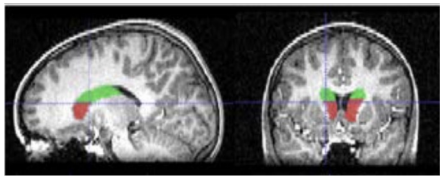
3D Reconstruction

Autonomous Navigation

Security Applications

Performance Analysis

Quality Control



Biomedical imaging for organ segment.

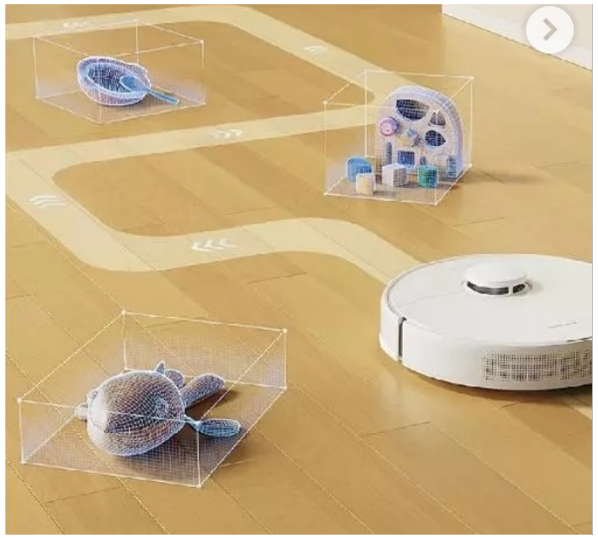
Vision?

Low level

Well defined extraction of features, simple images

High level

Extraction of high level features such as animals



And for Mobile Robotics?

Camera calibration

Estimates the parameters of the lens to correct for distortions, used to measure the size or displacement of objects in world from the camera scene

Camera localisation (or extrinsic calibration)

Estimate the pose of the camera in the real world

Object pose estimation and tracking

Estimate and track the position of known or moving objects in the image

3D reconstruction (stereo vision, structured light, …)

From successive images taken from different points of view, or with different lights that are known, estimate the distance to the different objects.

SLAM (simultaneous localisation and mapping)

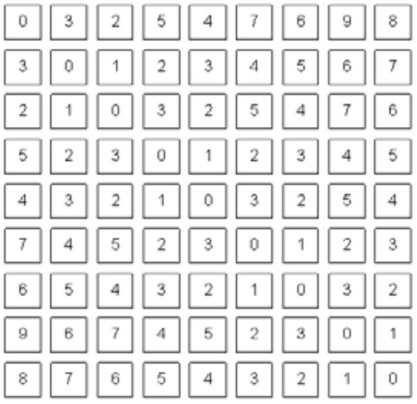
From unknown map and unknown camera poses, progressively construct the map of the environment while navigating through it

Digital Cameras

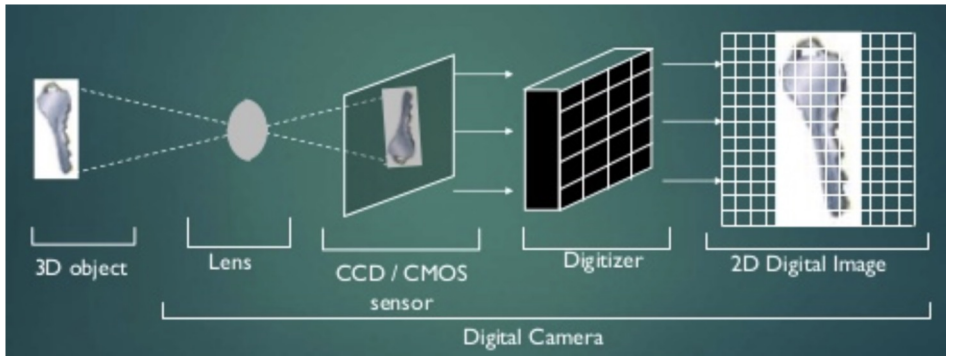
Digital cameras are everywhere. Why?

- Cheap in fabrication cost
- Cheap in energy consumption
- Versatile (multiple modalities, projections)
- Provide rich information on the environment
- Passive sensor
- Fast (some well over 250 fps)

What an old camera “sees”



What the computer sees



From an Image to Real World Information

To understand the scene, computer vision can rely on

2D aspects: known 2D patterns to understand the image, all the while considering that a good feature needs to be easy to extract and easy to match.

- Low-level features (geometric primitives) like points, lines, circles
- Higher-level features (objects recognition) like objects or places

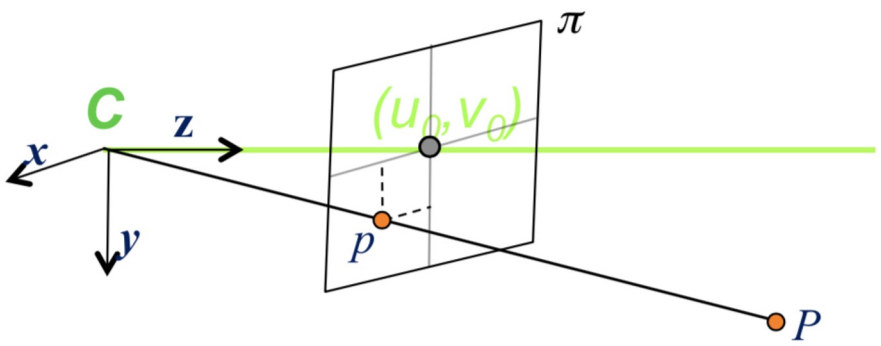
3D aspects: use our knowledge about the image formation (optics, scene…) to come back to 3D information

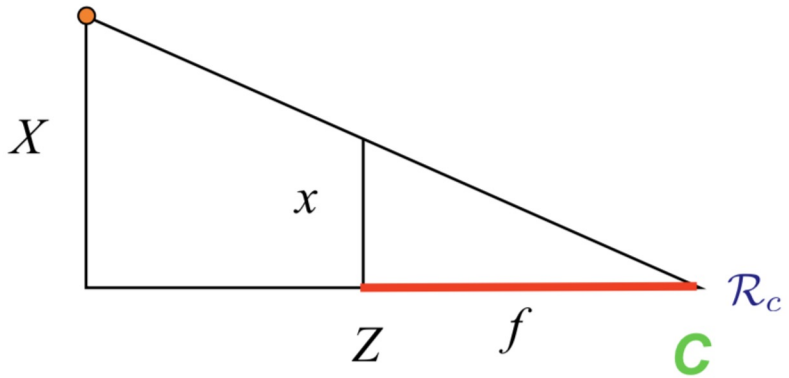
- Camera calibration
- Camera localization
- Object pose estimation/tracking
- 3D reconstruction
- SLAM (simultaneous localization and mapping)

From 3D to 2D – Perspective Projection

Definitions

- C is the center of projection
- **x** axes is parallel to image lines and **y** axes is parallel to image columns
- Intersection of **z** axes with image plane π is the principal point u_0, v_0
- focal distance $f = d(C,\pi)$





Credit: Eric Marchand INRIA Rennes

From 3D to 2D - Perspective Projection

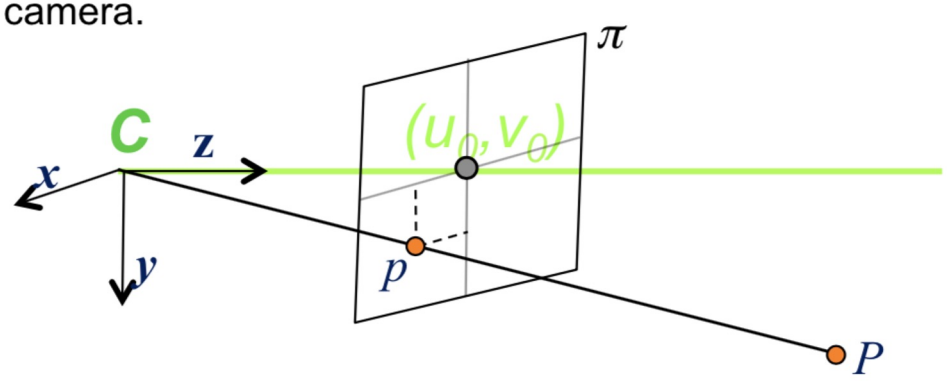
The perspective projection of a point X = *(X,Y,Z)* is the image point x = *(x,y)* with:

$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z}$$

All the points on the CP straight line given by:

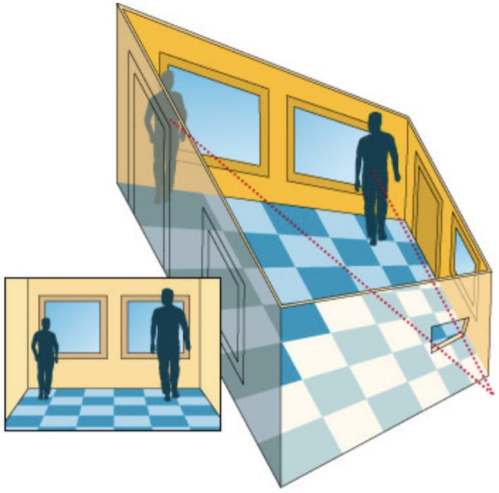
$$\begin{cases} fX - Zx = 0 \\ fY - Zy = 0 \end{cases}$$

are projected on the same point x. It is impossible to determine the position of P from p without *a priori* knowledge with one camera.



Credit: Eric Marchand INRIA Rennes

From 3D to 2D - Perspective Projection



2D Feature Processing

1. **Filtering to reduce noise in the image.**
2. **Preprocessing / Feature extraction**
    a. Edge detection
3. **Feature Detection**
    a. Hough Transform
    b. Pattern Matching
    c. Interest Point Detectors

or

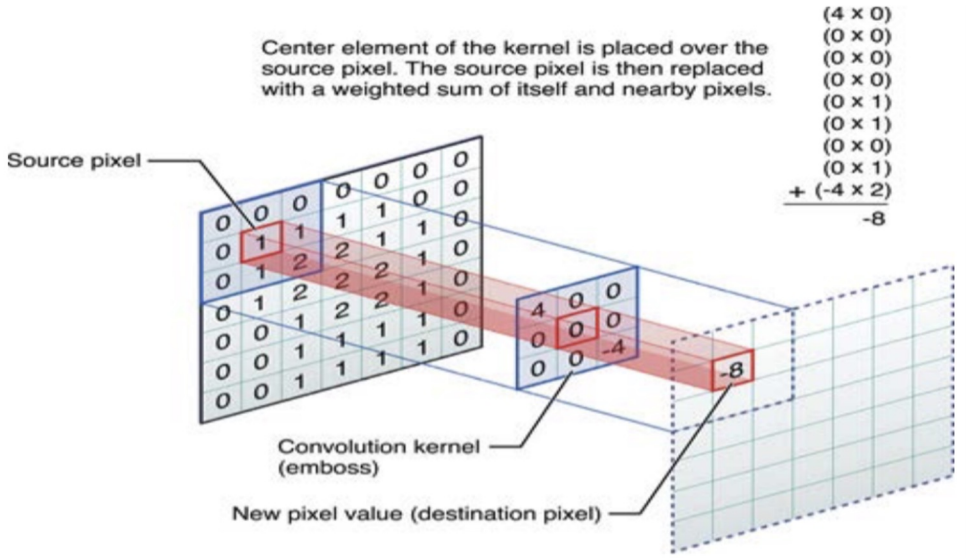
1. **Machine Learning** to train an algorithm to identify a particular feature

Filtering and Smoothing to get a good image

Gaussian smoothing

- Removes high-frequency noise
- Convolution* of intensity image I with a Gaussian kernel G:

$$\hat{I} = G \otimes I \qquad G = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

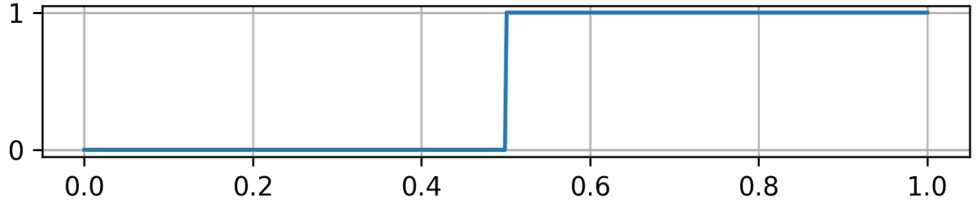


A convolution is an integral that expresses the amount of overlap of one function g as it is shifted over another function f:

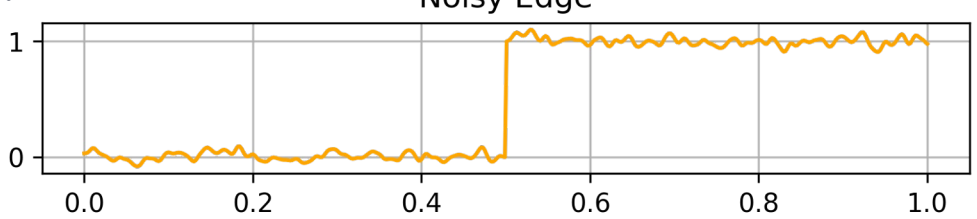
$$f \otimes g \equiv \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau = \int_{-\infty}^{\infty} g(\tau)\, f(t-\tau)\, d\tau$$

Filtering and Smoothing to get a good edge

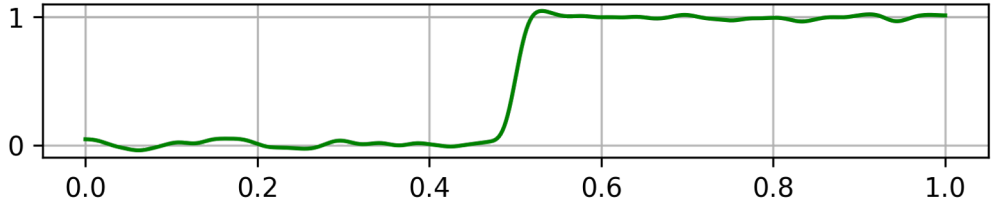
a.



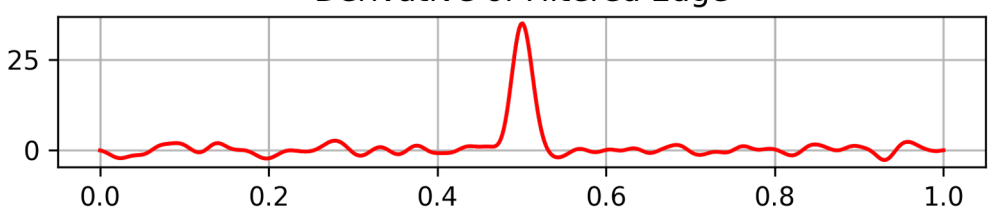
b.



c.



d.



a. Ideal 1D edge

Steps from image to edge detection:

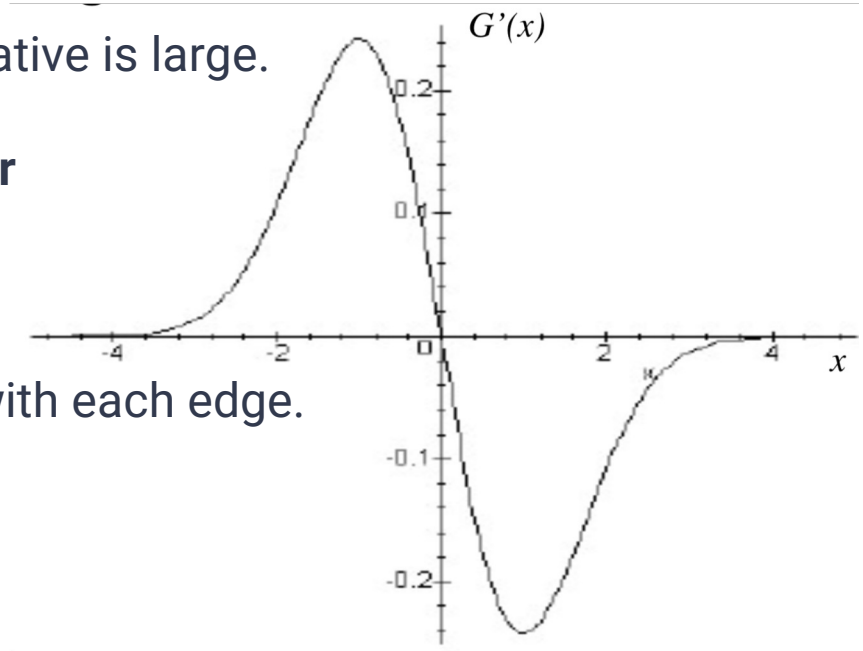
b. Noisy (realistic) edge
c. Filtered edge
d. Derivative of the image -> max=edge

Edge Detection (or Enhancement)

Goal: spot locations where the brightness undergoes sharp changes.

After filtering the image :

1. Differentiate one or two times the image.
2. Look for places where the magnitude of the derivative is large.



Example : the Canny Edge Filter an optimal edge detector with respect to the following criteria

- Maximizing signal to noise ratio.
- Achieving highest precision on the edge location.
- Minimizing the number of edge responses associated with each edge.

The processing steps :

- Convolution of the image with a Gaussian kernel G.
- Compute the derivative to find high gradient zones.
- Find maxima in the direction normal to the edge (non-maxima suppression).

Gaussian filtering and derivation can be combined into one operation: $(G \otimes I)' = G' \otimes I$

Edge Detection – Discrete Gradient Operators

Goal: numerically approximate the behavior of the Canny edge detector while reducing the computational cost and, in some cases, computing the edge orientation.

Note: G is the gradient magnitude and "Theta" the gradient direction

- **Roberts**

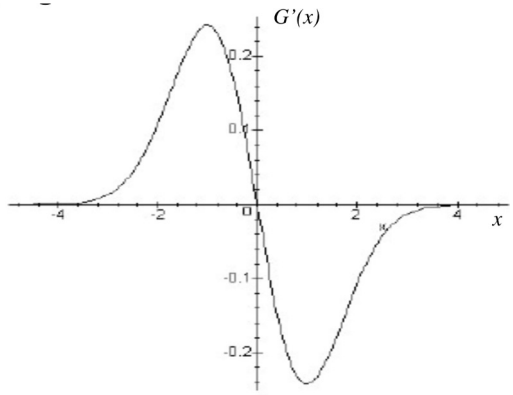
$$|G| \cong \sqrt{r_1^2 + r_2^2} \; ; \quad r_{1_m} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \; ; \quad r_{2_m} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

- **Prewitt**

$$|G| \cong \sqrt{p_1^2 + p_2^2} \; ; \quad \theta \cong \operatorname{atan}\left(\frac{p_1}{p_2}\right) ; \quad p_{1_m} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \; ; \quad p_{2_m} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

- **Sobel**

$$|G| \cong \sqrt{s_1^2 + s_2^2} \; ; \quad \theta \cong \operatorname{atan}\left(\frac{s_1}{s_2}\right) ; \quad s_{1_m} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \; ; \quad s_{2_m} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Edge Detection – Discrete Gradient Operators

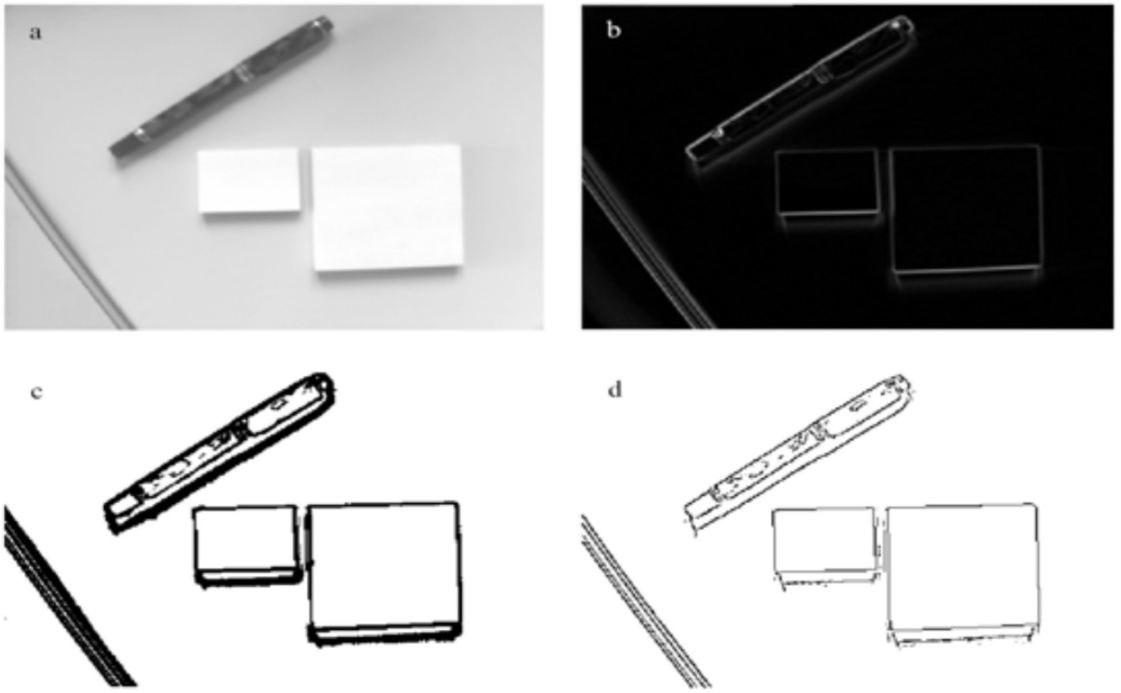
Example of the Sobel Filter

a. Raw image

b. Filtered (Gaussian or Sobel)

c. Thresholding

d. Nonmaxima suppression



Feature Detection – Hough Transform

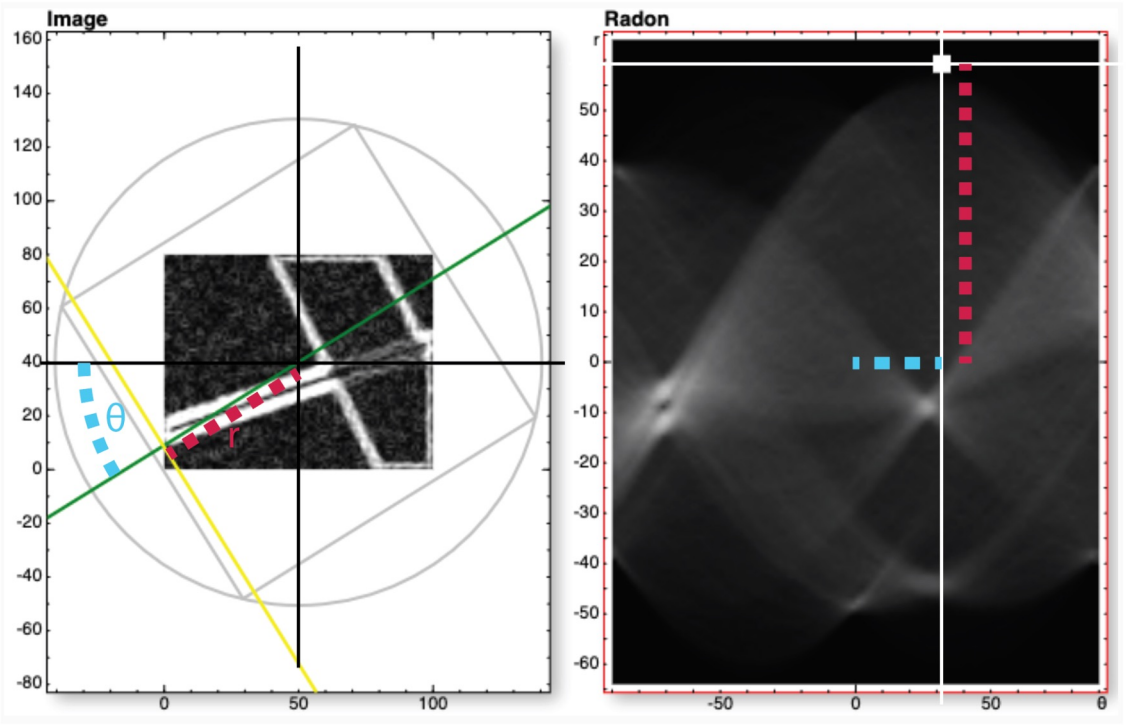
Goal: extract straight edges, which are often used in mobile robotics (e.g. doors, lane).

Each line in the image can be parameterized as:

$$\rho(\theta) = x\cos\theta + y\sin\theta$$

The Hough transform is a 2D histogram in the parameter space (ρ,θ) whose cells count the number of *edge pixels* that a particular line is voting for.

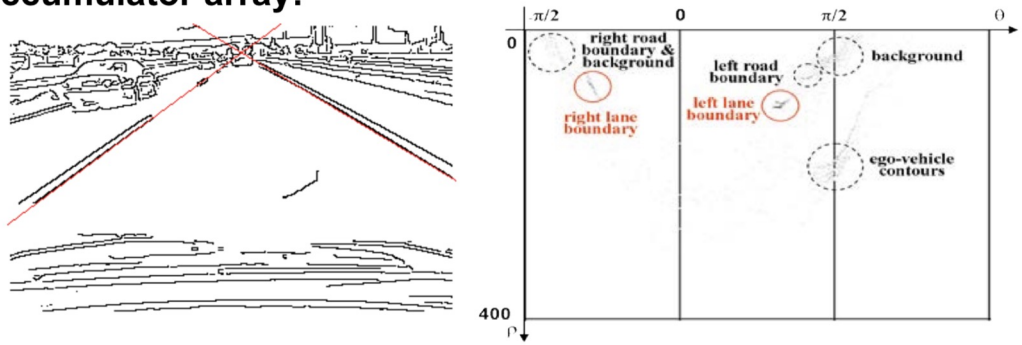
Note: the Hough transformation is assuming a preprocessing step for edge detection.



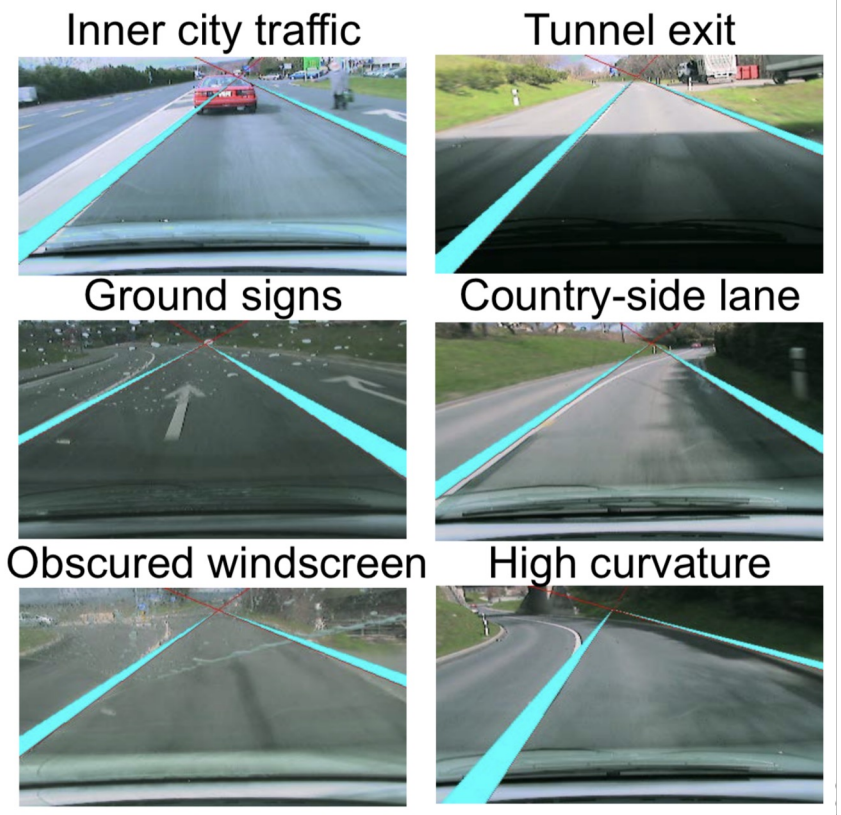
Note: the Hough transform can also be used to detect other shapes than lines depending on the used parameterization.

Feature Detection – Hough Transform

A Canny edge pixel map and the resulting Hough transform accumulator array:



Lane boundaries: $(\theta_l, \rho_l) = \{59°, 105\}$, $(\theta_r, \rho_r) = \{-51°, 78\}$



Feature Detection – Pattern Matching

Finding a known pattern in the input, e.g. an image template *I1* in a (larger) image *I2*

Can be computed by either the :

- sum of squared differences which should be minimised
- correlation which should be maximised

Applications :

- **Object recognition & tracking** : searching for the stored patterns in the inputs. Peaks indicate the object position in the image
- **Optic flow :** Match patches of previous image with new image. Normalising the peak by the time difference gives the optic flow
- **Stereo vision :** Match patches of the left image with the right image. Peak corresponds to disparity in the position of the patch

Feature Detection – Pattern Matching

- sum of squared differences:

$$SSD(D_x, D_y) = \sum_{(x,y)} (I_1(x,y) - I_2(x + D_x, y + D_y))^2$$

- correlation:

$$Correlation(D_x, D_y) = \sum_{(x,y)} I_1(x,y) \cdot I_2(x + D_x, y + D_y)$$

where $I(x,y)$ is the irradiance* of the image point (x,y), and D_x, D_y are the translation components of the template I_1 across the image I_2

Correlation can be computed faster via Fast Fourier transform:

$$I_1 * I_2 = FFT^{-1}(FFT(I_1) \cdot \overline{FFT(I_2)})$$

* mean should be substracted

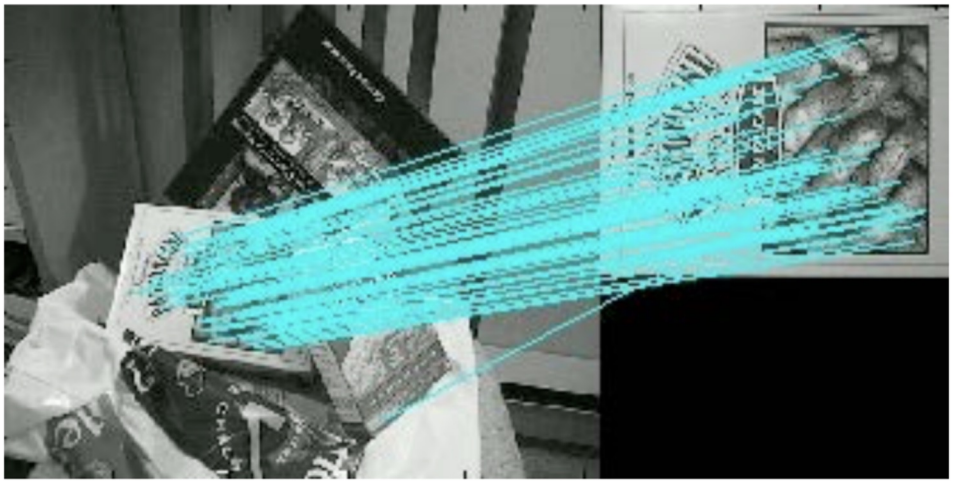
Feature Detection – Interest Point Detectors

Interest point detectors are fundamental for many problems in computer vision:

- object recognition from various perspectives
- 3D structure from multiple images (stereo / optic flow)
- motion tracking

Ideal features (interest points) should be invariant to

- image scaling and rotation
- change in illumination
- camera viewpoint
- highly distinctive (to enable correct match with high probability against a large database of features)



http://people.cs.ubc.ca/~lowe/keypoints/

Feature Detection – Interest Point Detectors

	Corner detector	"Blob" detector	Rotation invariant	Scale invariant	Affine invariant	"Repeatability"	Localization accuracy	Robustness	Computational efficiency
Harris (Harris et al., 1988)	x		x			+++	+++	++	++
Harris-Laplacian (Mikolajczyk et al., 2004)	x	x	x	x		+++	+++	++	+
Harris-Affine (Mikolajczyk et al., 2002)	x	x	x	x	x	+++	+++	++	++
SUSAN (Smith et al., 1997)	x		x			++	++	++	+++
FAST (Rosten et al., 2005)	x		x			++	++	++	++++
SIFT (Lowe et al., 1999)		x	x	x	x	+++	++	+++	+
MSER (Matas et al., 2002)		x	x	x	x	+++	+	+++	+++
SURF (Bay et al., 2008)		x	x	x	x	++	++	++	++

Feature Detection – SIFT (Scale-Invariant Feature Transform)

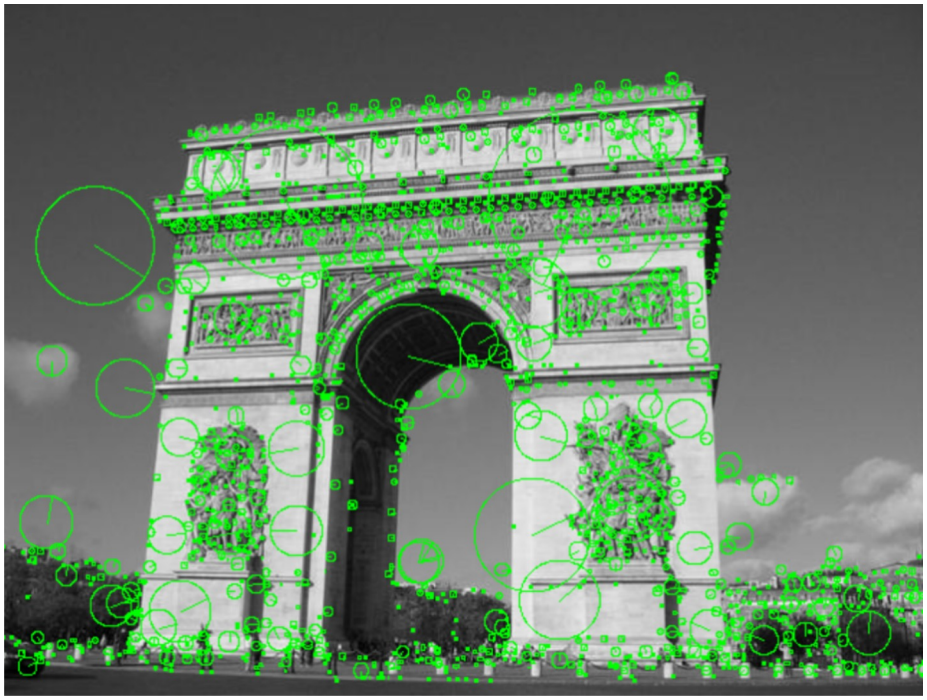
SIFT detects a series of keypoints from a **multiscale image representation**. This multiscale representation consists of a family of increasingly blurred images. Each keypoint is a blob-like structure whose **center position (x, y)** and **characteristic scale σ** are accurately located. SIFT computes the **dominant orientation θ** over a region surrounding each one of these keypoints. For each keypoint, the **quadruple (x, y, σ, θ) defines the center, size and orientation of a *normalized* patch** where the SIFT descriptor is computed. As a result of this normalization, *SIFT keypoint descriptors are in theory invariant to any translation, rotation and scale change*. The descriptor encodes the spatial gradient distribution around a keypoint by a 128-dimensional vector. This feature vector is generally used to match keypoints extracted from different images.

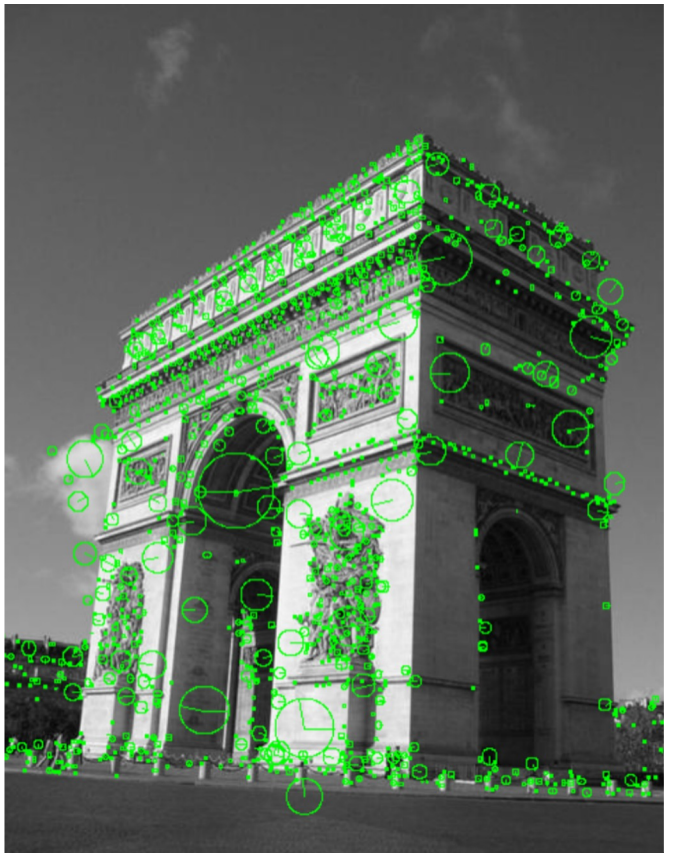
Rey Otero & Ives and Delbracio (2014) Anatomy of the SIFT Method, Image Processing On Line, 4 pp 370--396, 2014, https://doi.org/10.5201/ipol.2014.82

Feature Detection – SIFT (Scale-Invariant Feature Transform)

Rey Otero & Ives and Delbracio (2014) Anatomy of the SIFT Method, Image Processing On Line, 4 pp 370--396, 2014, https://doi.org/10.5201/ipol.2014.82

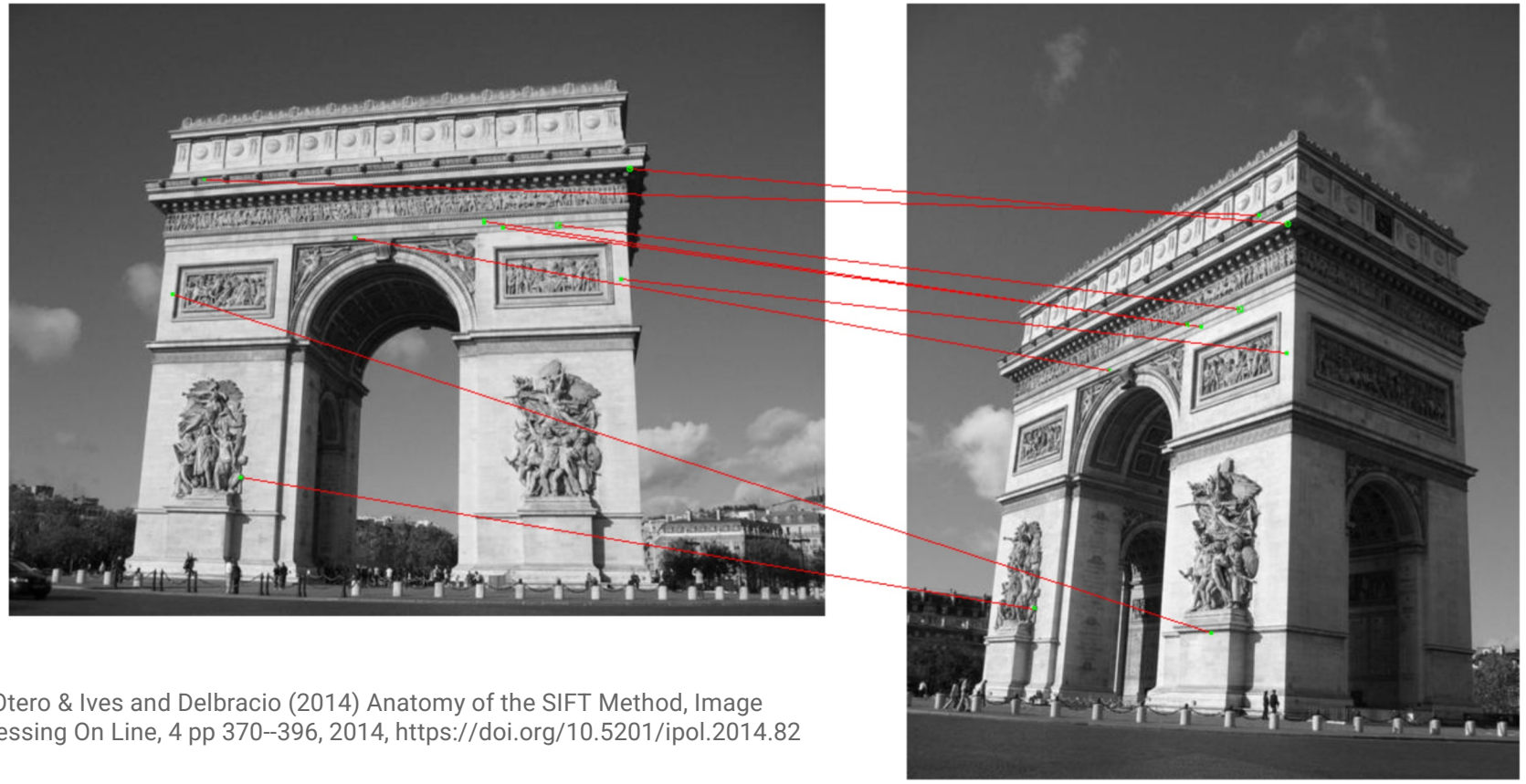
Feature Detection – SIFT (Scale-Invariant Feature Transform)





Rey Otero & Ives and Delbracio (2014) Anatomy of the SIFT Method, Image Processing On Line, 4 pp 370--396, 2014, https://doi.org/10.5201/ipol.2014.82

Feature Detection – SIFT (Scale-Invariant Feature Transform)

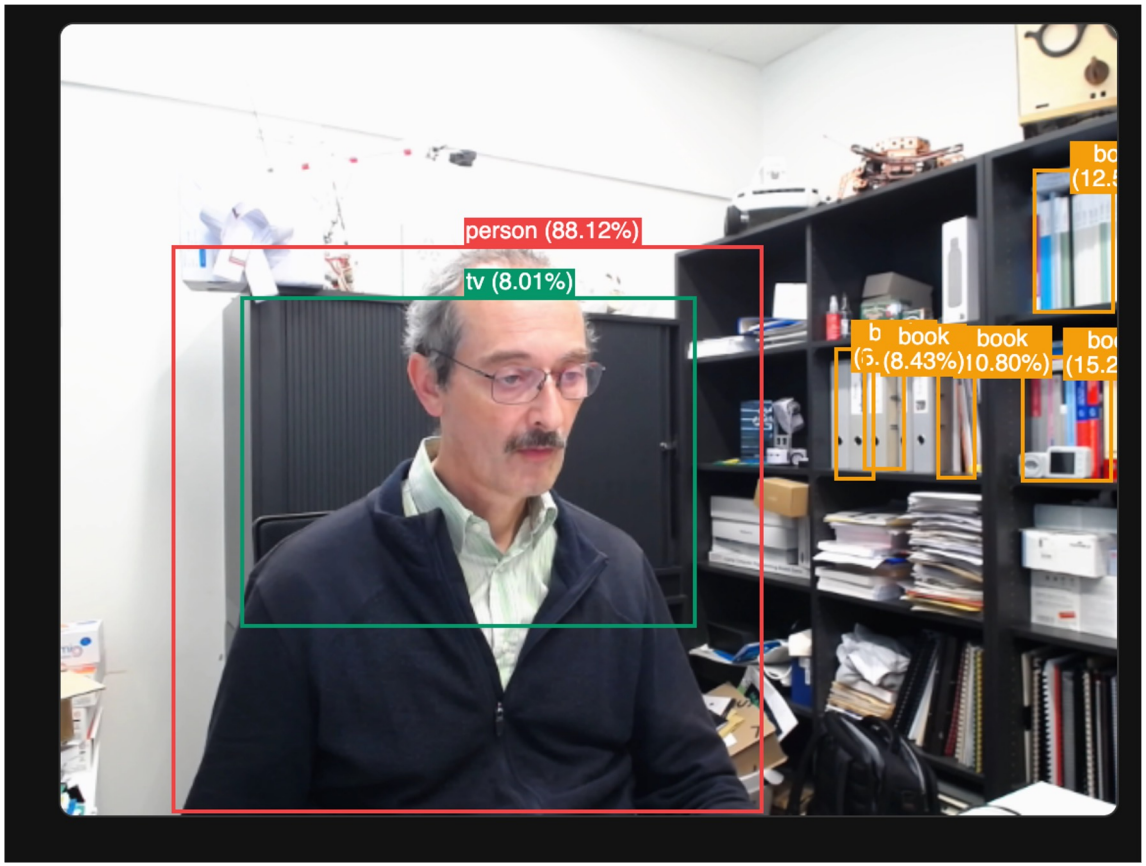


Rey Otero & Ives and Delbracio (2014) Anatomy of the SIFT Method, Image Processing On Line, 4 pp 370--396, 2014, https://doi.org/10.5201/ipol.2014.82

What should I remember

- Why vision is everywhere
- 2D vs 3D aspects
- Projection of 3D on 2D
- Canny Edge Filter
  - Combination of smoothing and edge detection into one operator
  - Numerical approximations
- Hough Transform
  - concept
  - reading a graph
- Pattern matching
  - concept
  - implementations by sum of squared differences or correlation
- Interest point detectors
  - Basic concept, knowing there are many implementations with different features

“Artificial intelligence”: object detection

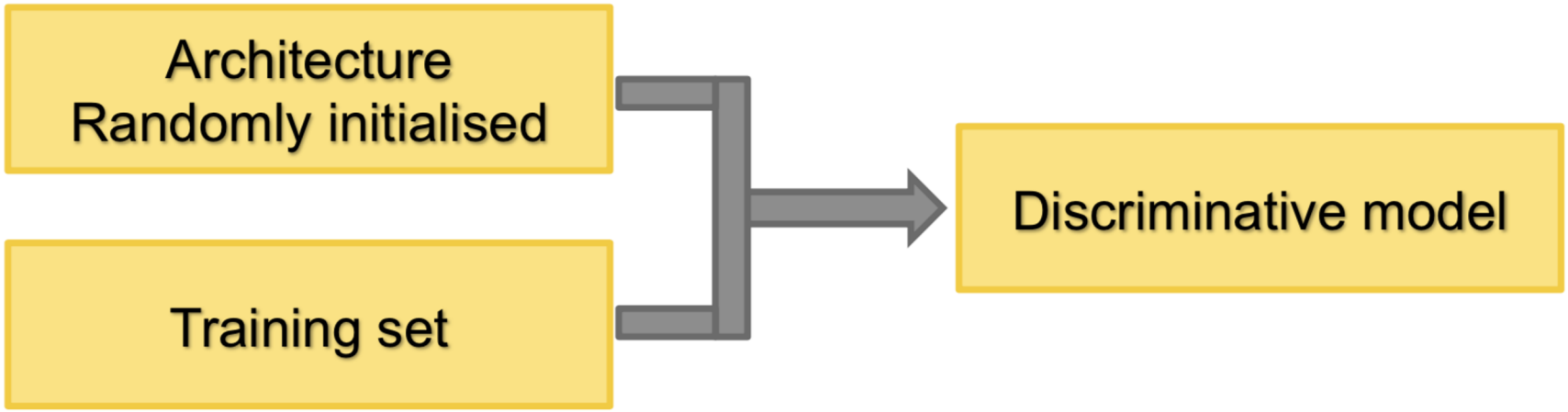


https://huggingface.co/spaces/saraivaai/detecao-objeto

How can you recognize a person?

Machine Learning

Before: All of the models were hand-crafted: researchers defined approaches to discriminate one pattern from another. Works well for “simple” scenes, but in the wild they hardly work as appearances can have many variations.
Now: many labeled images are available, it is possible to learn those variations and their discriminative models, which corresponds to finding the optimal parameters for one parametric model (architecture).



The Architecture is not random …..

object detection

DREAME

Community Submit a request Sign in

Dreame Tech > Troubleshooting > Robot Vacuums > Troubleshooting For X40 Master

Search

Articles in this section

- X40 Master: How to Fix an Unstable Mop Pad Holder
- How to Power Off Your X40 Master Robot
- X40 Master Guide: Solving Base Station Return and Location Struggles
- Resolving Network Distribution Issues with the X40 Master
- Resolving Sewage Tank and Cleaning Pan Issues with the X40 Master
- Tackling Abnormal Dust Collection with the X40 Master
- Fixing Breakpoint Resume Issues with the X40 Master
- How to Deal with Foreign

Enhancing Obstacle Recognition with Your X40 Master

Customer Support Specialist
9 months ago · Updated

Follow

🤖 Is Your X40 Master Struggling to Recognize Obstacles? 🤖

If your robot vacuum's AI is having trouble identifying obstacles correctly, here are some steps to help you troubleshoot and improve its performance:

---

1. Startup Function:

- Ensure that the "AI Intelligent Recognition" feature is activated on your robot. This is the first step to leveraging the AI's obstacle recognition capabilities.

---

2. Recognition Range:

- Understand that the AI's intelligent recognition currently covers items like "shoes" (including slippers), "socks" (and rags), "wires", "scales", and "bases".

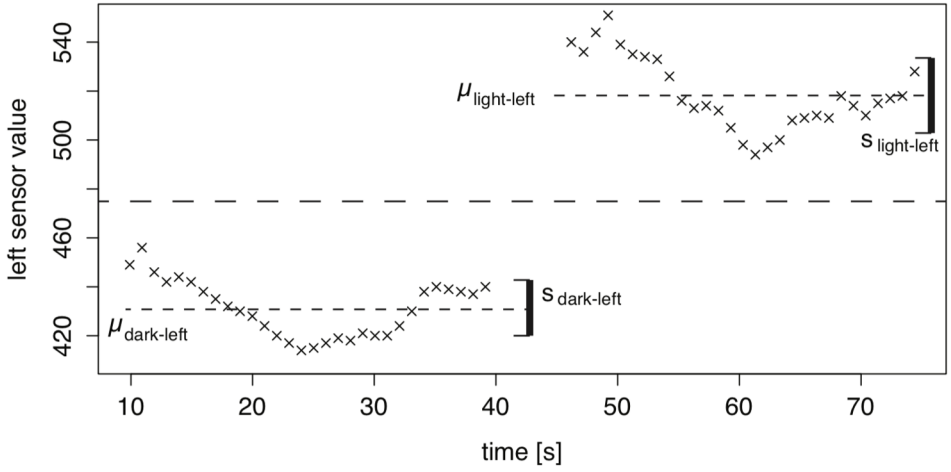
Machine Learning

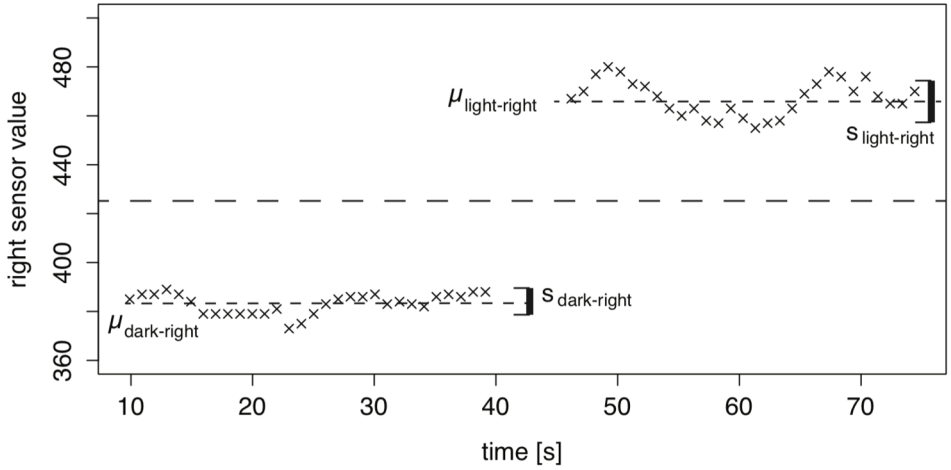
Statistical analysis
(specific transparent model)

Machine learning
(generic black-box model)

Machine Learning : Linear Discriminant

How would we proceed to distinguish between two colors? Try to find a discriminative rule between the sensor values of both colors. This criteria ideally separates well between the elements of different classes (the two colors) minimizes the spread within a given class.





Machine Learning : Linear Discriminant

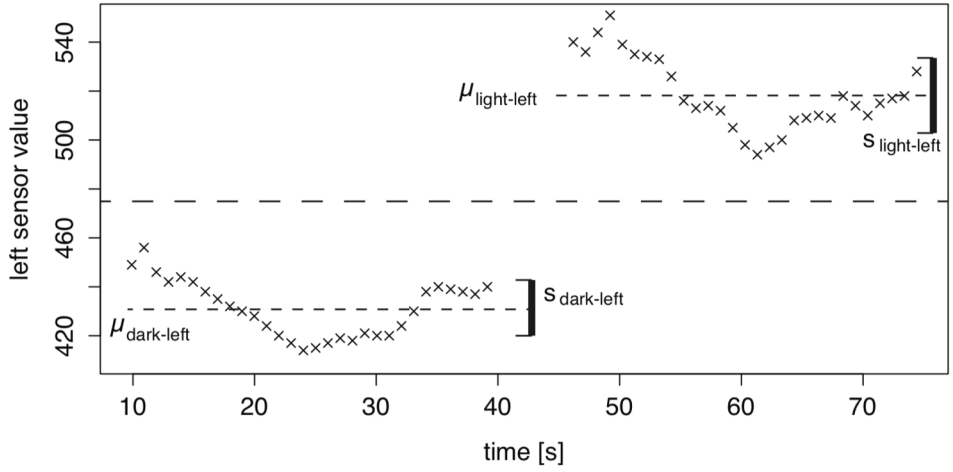
How would we proceed to distinguish between two colors?
Try to find a discriminative rule between the sensor values of both colors. This criteria ideally separates well between the elements of different classes (the two colors) minimizes the spread within a given class.

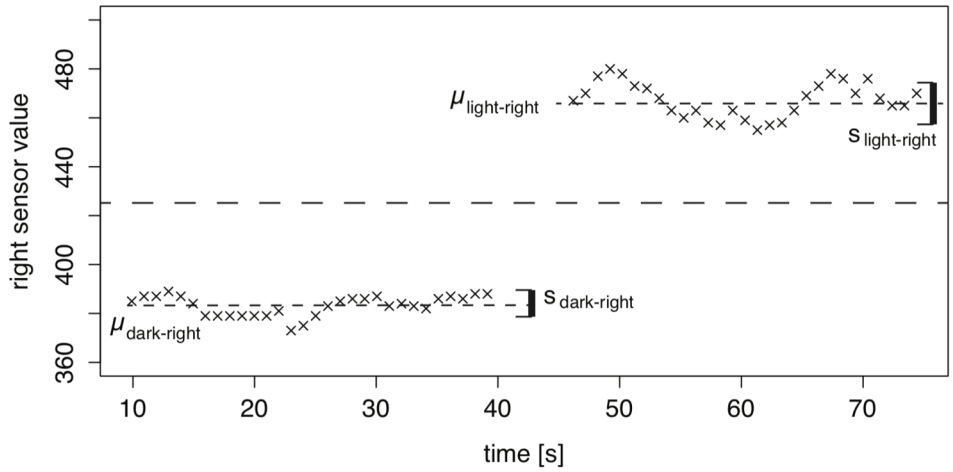
Discriminator:

$$\Delta = \frac{\mu_{dark} + \mu_{light}}{2}$$

Quality criteria:

$$J_k = \frac{\left(\mu^k_{dark} - \mu^k_{light}\right)^2}{\left(s^k_{dark}\right)^2 + \left(s^k_{light}\right)^2}$$





Machine Learning : Linear Discriminant

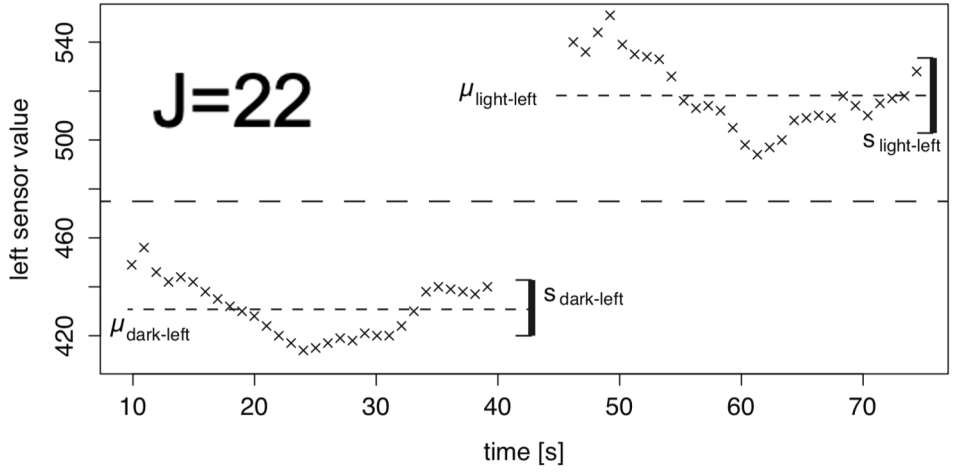
How would we proceed to distinguish between two colors?
Try to find a discriminative rule between the sensor values of both colors. This criteria ideally separates well between the elements of different classes (the two colors) minimizes the spread within a given class.

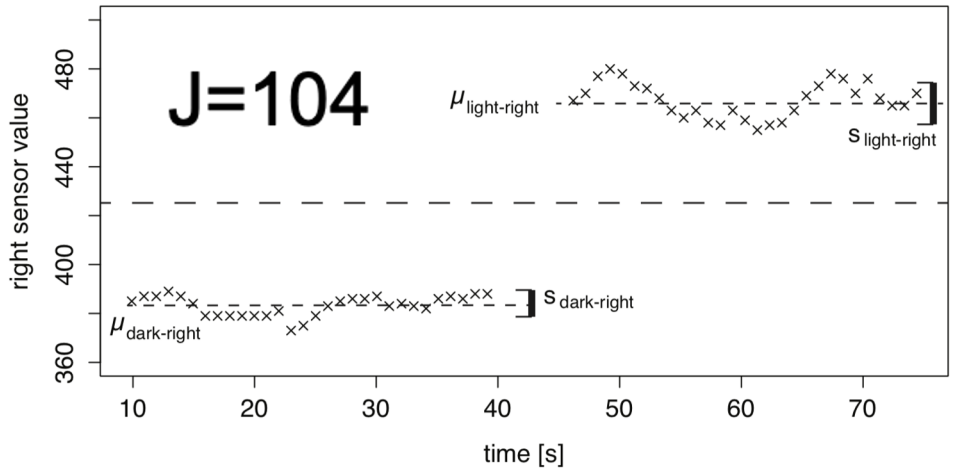
Discriminator:

$$\Delta = \frac{\mu_{dark} + \mu_{light}}{2}$$

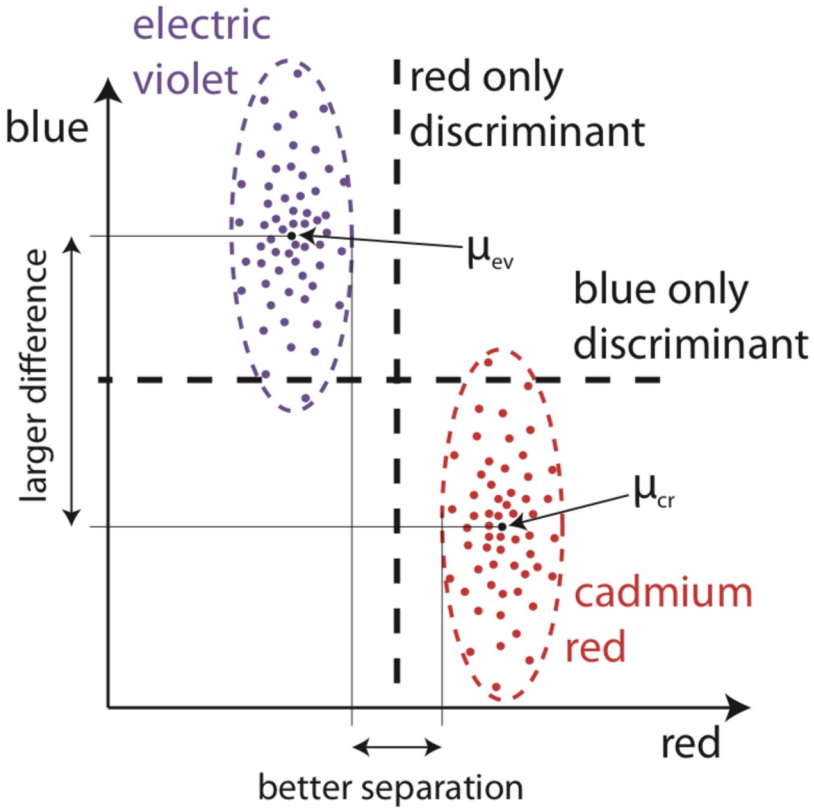
Quality criteria:

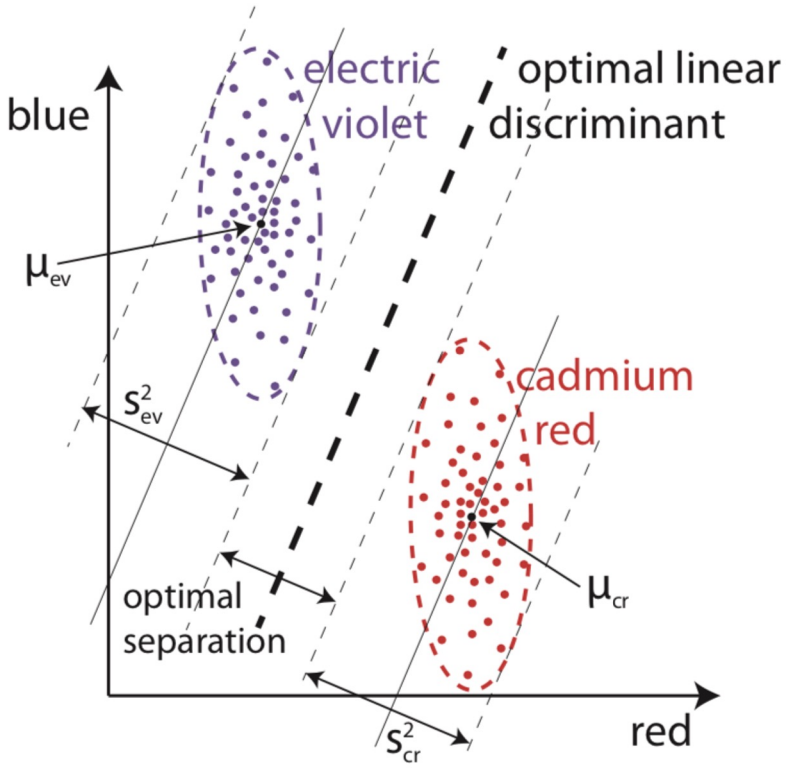
$$J_k = \frac{\left(\mu^k_{dark} - \mu^k_{light}\right)^2}{\left(s^k_{dark}\right)^2 + \left(s^k_{light}\right)^2}$$



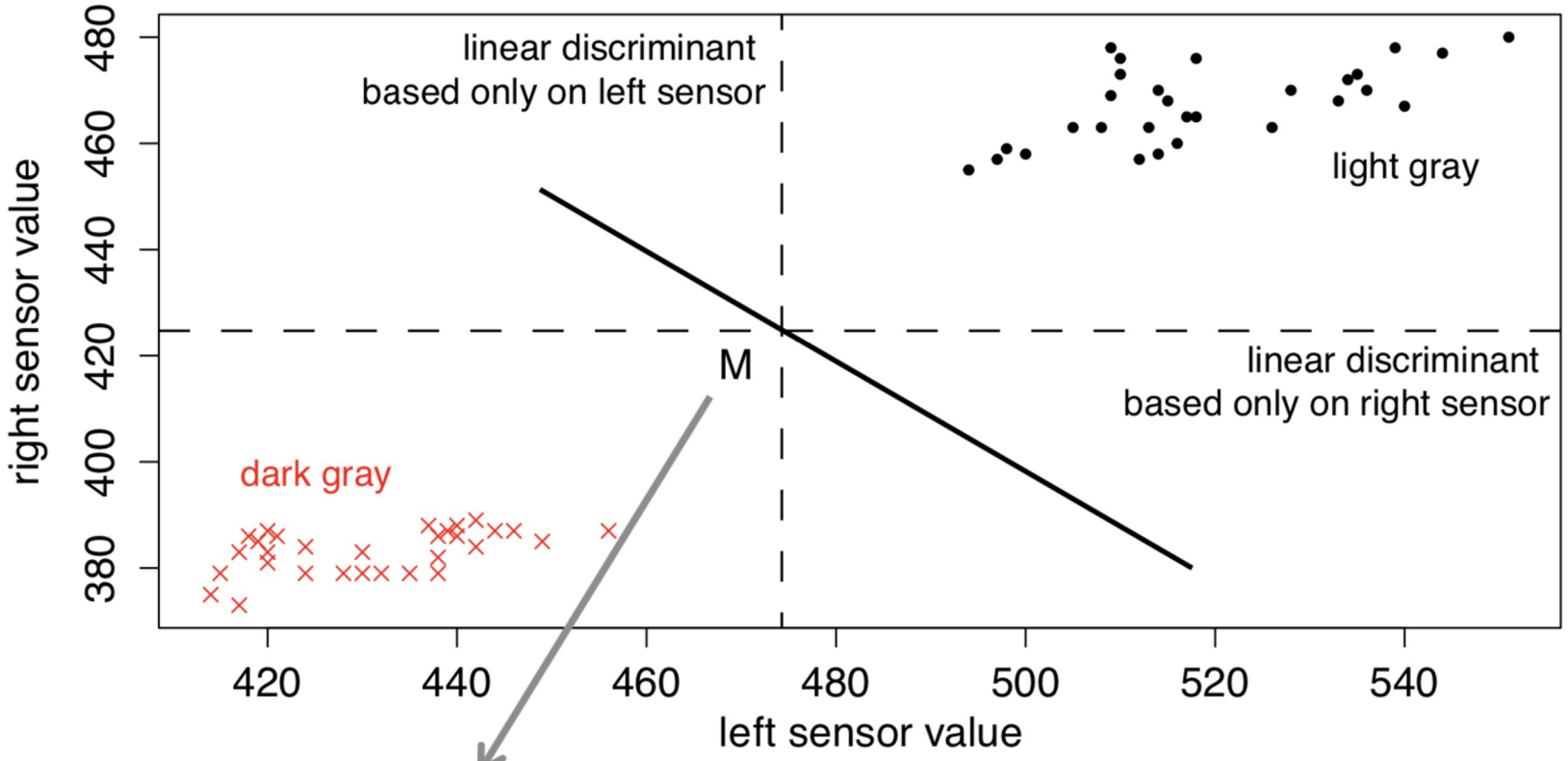


Machine Learning : Linear Discriminant





Machine Learning : Linear Discriminant



$$\left(\frac{\mu_{light}^{left} + \mu_{dark}^{left}}{2}, \frac{\mu_{light}^{right} + \mu_{dark}^{right}}{2} \right)$$

Machine Learning : Linear Discriminant Slope

How to choose the slope? The slope of the discriminant line should maximise the quality criteria J.

The discriminant line will have the form:

$$w_1 x_1 + w_2 x_2 = c$$

where x_1 and x_2 are the two inputs, w_1 and w_2 define the slope and c is a constant ensuring that M is on the discriminant li

By derivation and finding the max (= 0) we get:

$$\mathbf{w} = \mathbf{S}^{-1} (\boldsymbol{\mu}_{light} - \boldsymbol{\mu}_{dark})$$

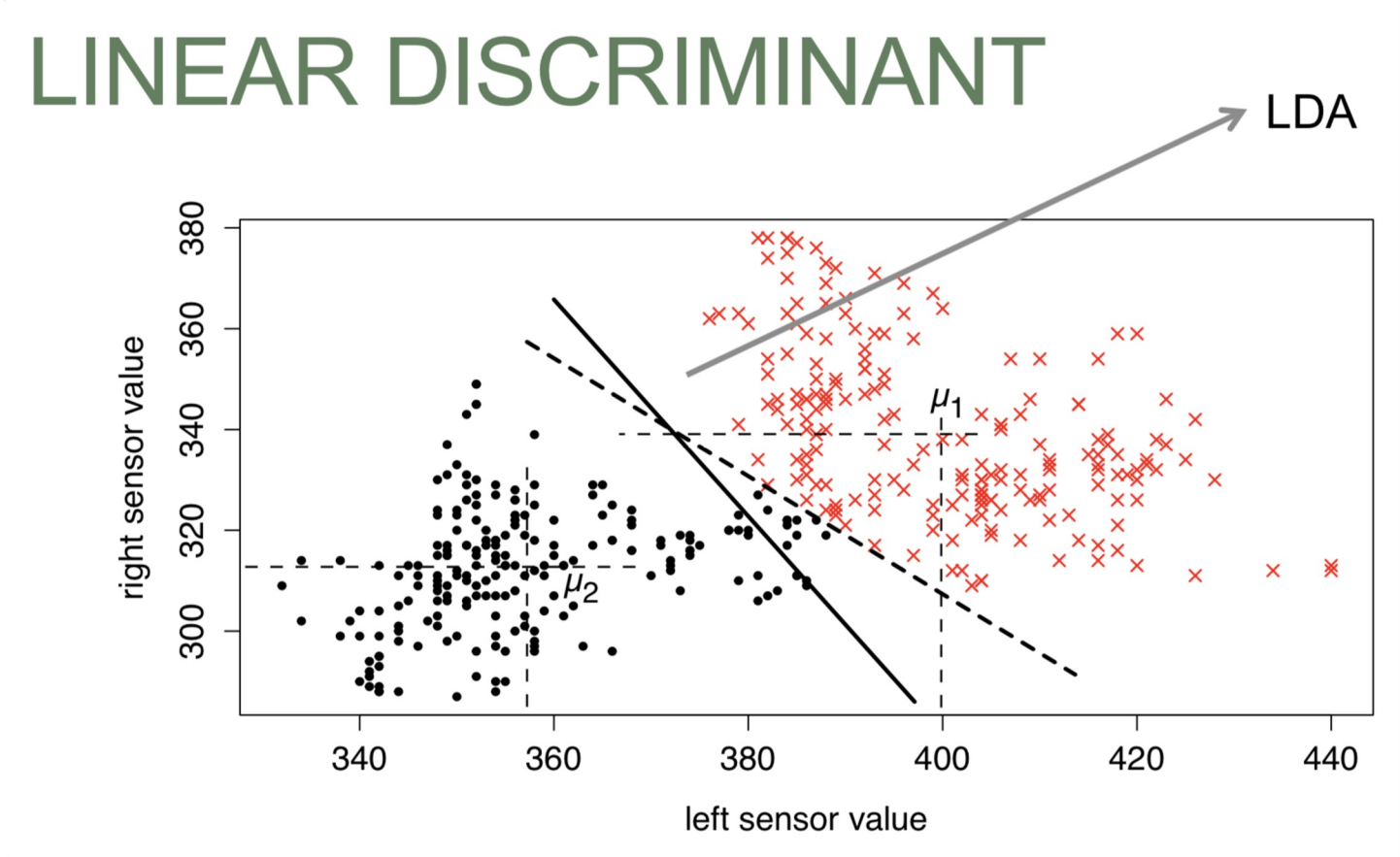
where:

$$\boldsymbol{\mu}_{light} = \begin{bmatrix} \mu_{light}^{left} \\ \mu_{light}^{right} \end{bmatrix}, \quad \boldsymbol{\mu}_{dark} = \begin{bmatrix} \mu_{dark}^{left} \\ \mu_{dark}^{right} \end{bmatrix}$$

are the mean vectors of the two classes and $\mathbf{S}^{-1}$ is the inverse of the average of the covariance matrices of the two classes[6]:

$$\mathbf{S} = \frac{1}{2} \left(\begin{bmatrix} s^2\left(\mathbf{x}_{light}^{left}\right) & cov\left(\mathbf{x}_{light}^{left}, \mathbf{x}_{light}^{right}\right) \\ cov\left(\mathbf{x}_{light}^{right}, \mathbf{x}_{light}^{left}\right) & s^2\left(\mathbf{x}_{light}^{right}\right) \end{bmatrix} + \begin{bmatrix} s^2\left(\mathbf{x}_{dark}^{left}\right) & cov\left(\mathbf{x}_{dark}^{left}, \mathbf{x}_{dark}^{right}\right) \\ cov\left(\mathbf{x}_{dark}^{right}, \mathbf{x}_{dark}^{left}\right) & s^2\left(\mathbf{x}_{dark}^{right}\right) \end{bmatrix} \right).$$

Machine Learning : Opt. Linear Discriminant



Beyond ML : Towards Artificial Neural Nets (ANN)

What if the robot could learn the same way a brain does? 10^{11} neurons..

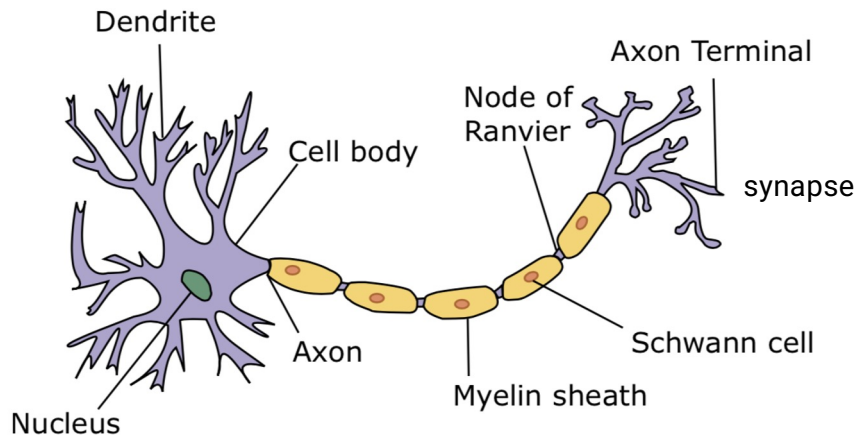


Fig. 13.1 Structure of a neuron

Real Neuron

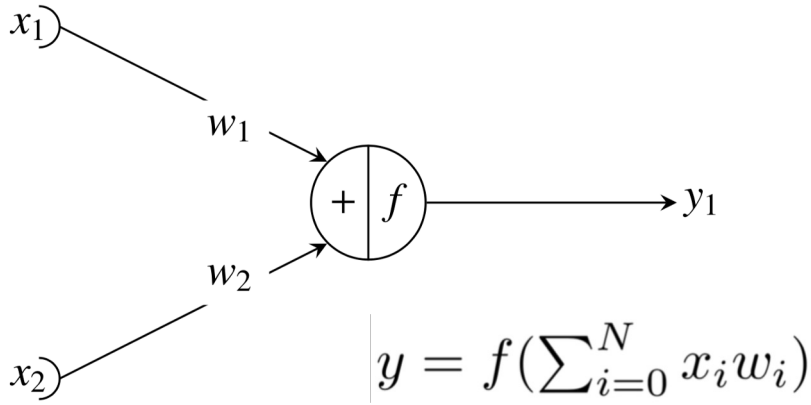


Fig. 13.2b ANN: one neuron with two inputs

Artificial Neuron

Beyond ML : ANN Output Function f

What type of output functions are typically used for ANNs?

$$y = f(\sum_{i=0}^{N} x_i w_i)$$

Function f:

- Step (binary):

  y = 0 if sum <= 0, y = 1 if sum > 0

- Saturation:

  y = 0 if sum <= 0, y = 1 if sum >= 1, y = sum otherwise

- Sigmoid function:

  $y = 1 / (1 + e^{-sum})$

Beyond ML : ANN Structure

In such a structure, a single neuron with a step function output can be seen as a classifier of the inputs into two classes (outputs 0 or 1). In the case of two inputs, the separation between the two classes is given by the following function:

$$x_1 w_1 + x_2 w_2 = 0$$

This is a linear separator with parameters w_1 and w_2. Often a third fixed input is added:

$$x_1 w_1 + x_2 w_2 + w_3 = 0$$

To implement non linear functions, create a structure with multiple layers.

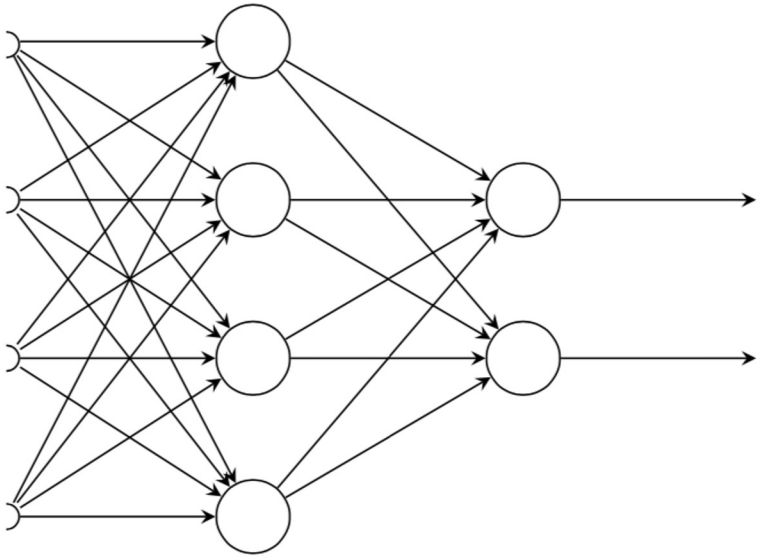


Fig. 13.6a Multilayer ANN

Beyond ML : ANN Structure

To add memory we use recurrent connections:

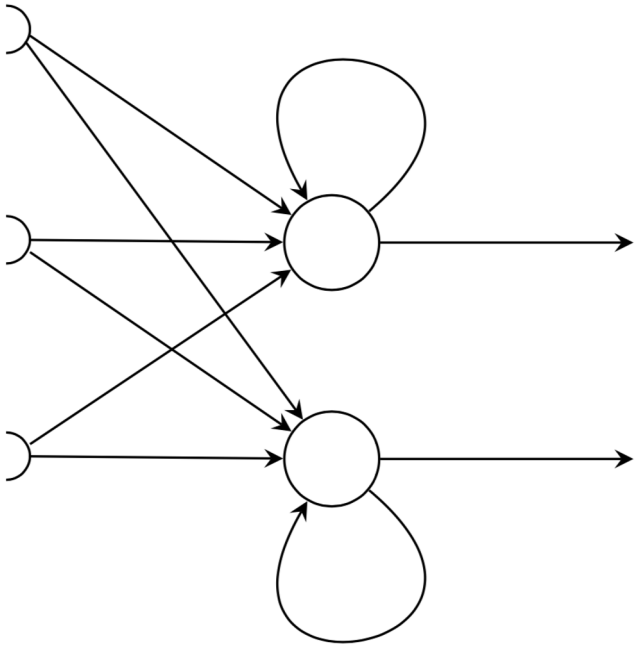


Fig. 13.6b ANN with memory

Beyond ML : ANN Learning

In ML, three types of learning algorithms exist :

• **Supervised learning :** when we know what output is expected for a set of inputs. Requires having a labelled dataset.

• **Unsupervised learning :** when the goal is to learn the underlying structure of the data without relying on the labels. In such cases, the network maps a large number of inputs

• **Reinforcement learning** : we simply tell the network if the output it computes is good or not. This is for instance done by the Hebb rule which positively reinforces behaviours which lead to a good output and “punishes” behaviours which lead to negative outputs.

When there is a good action:

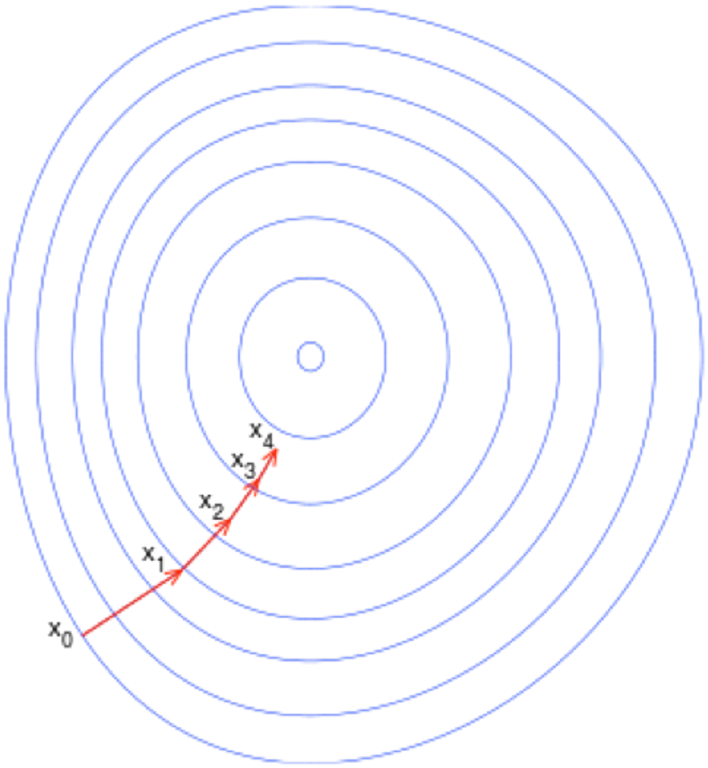
$$\Delta w_{kj} = \alpha \, x_k \, y_j$$

From ANNs to Deep Neural Nets : Supervised L.

Goal: find the set of parameters which optimize a cost function. In image labeling for instance, we want for each image as input, its label as output. The cost function is how far is our output from the desired label.

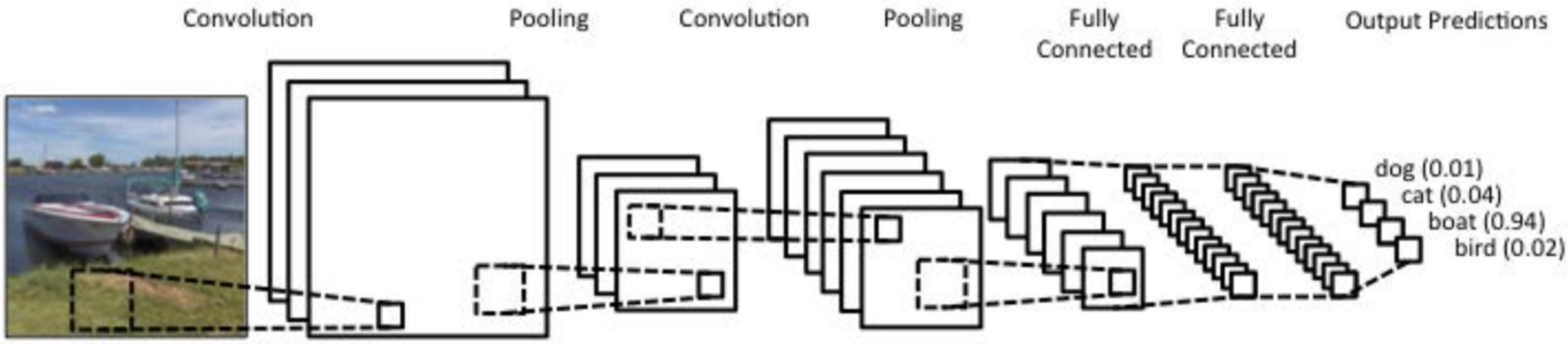
Common approach: the cost function derivatives (or slope) can be estimated for each set of parameters. The gradients can be followed to find the optimums.

Analogy: you are blind folded on a hill and you want to find the top. You can still feel the slope of the hill with your feet, how do you do ?



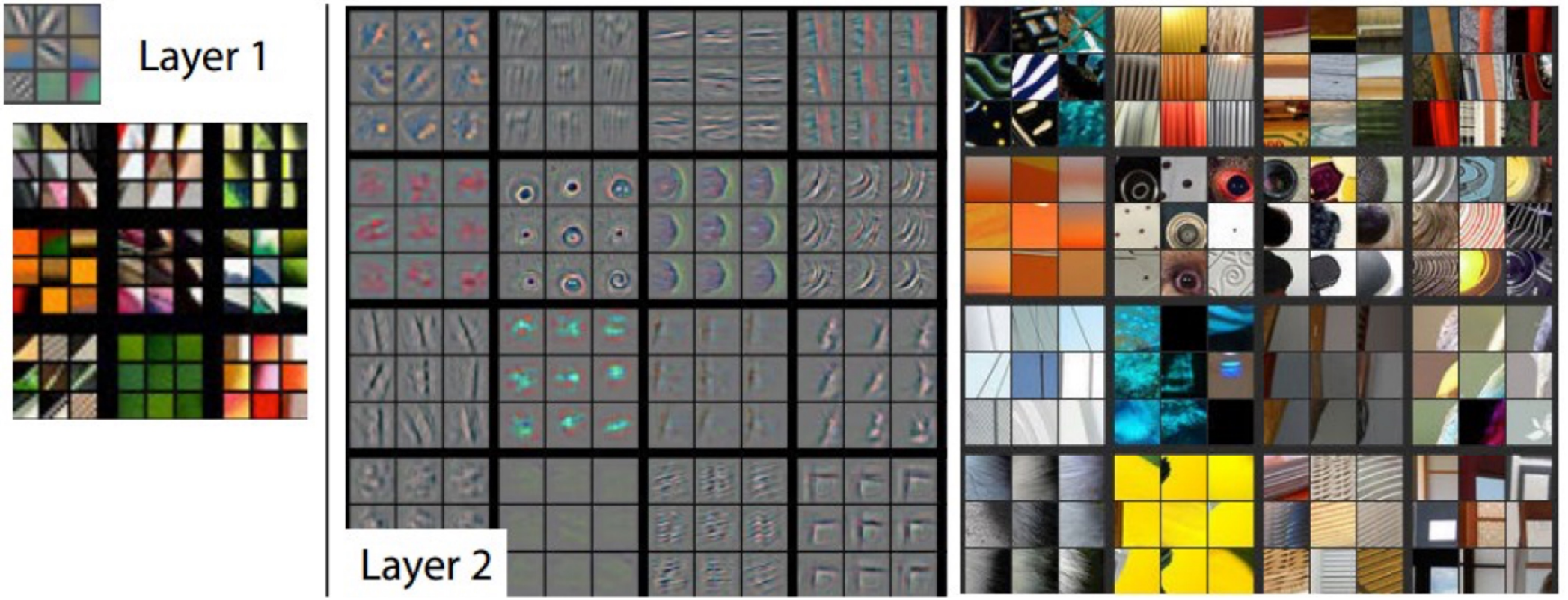
Beyond ML : From ANNs to Deep Neural Nets

Combining machine learning and deep learning with huge datasets through convolutional neural networks (CNNs) to achieve powerful results.



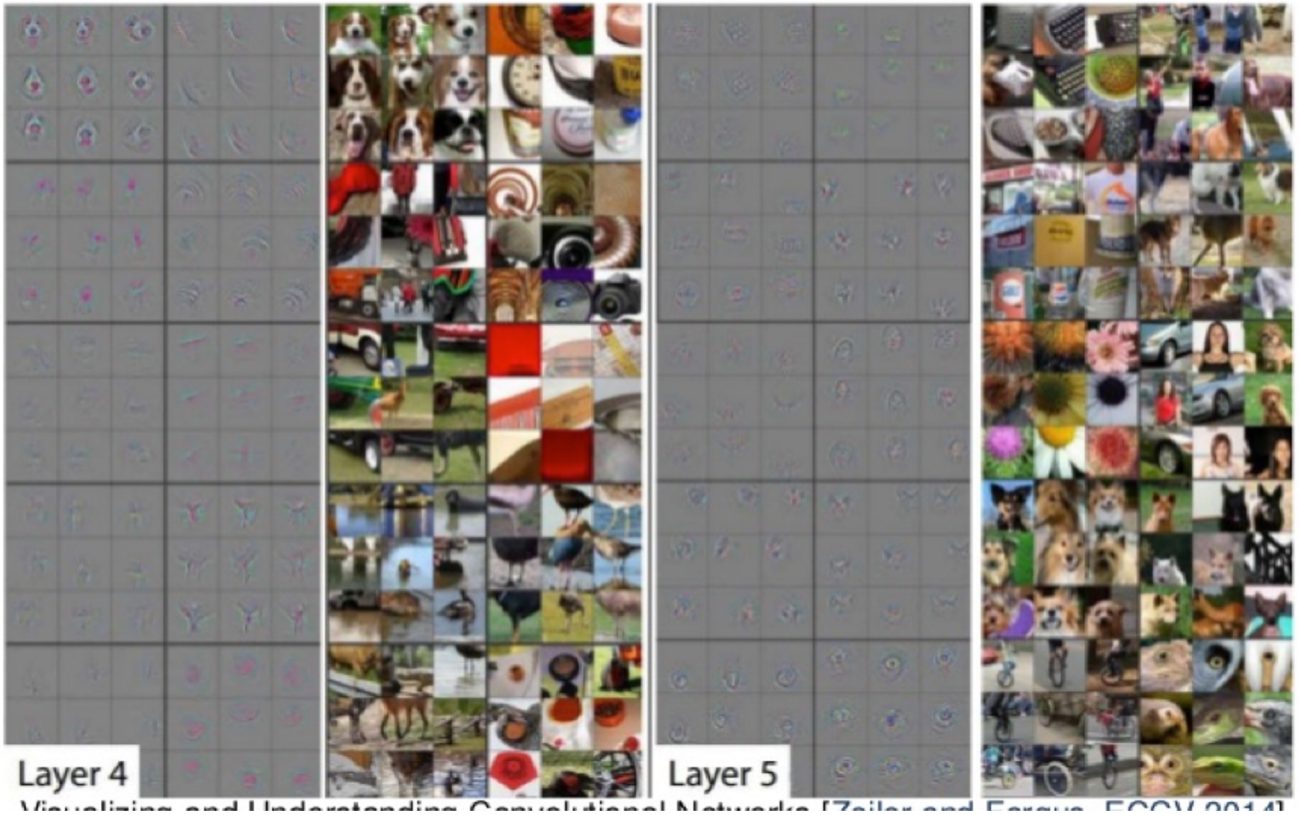
CNNs are stacked 2D convolution filters.

Beyond ML : From ANNs to Deep Neural Nets



Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

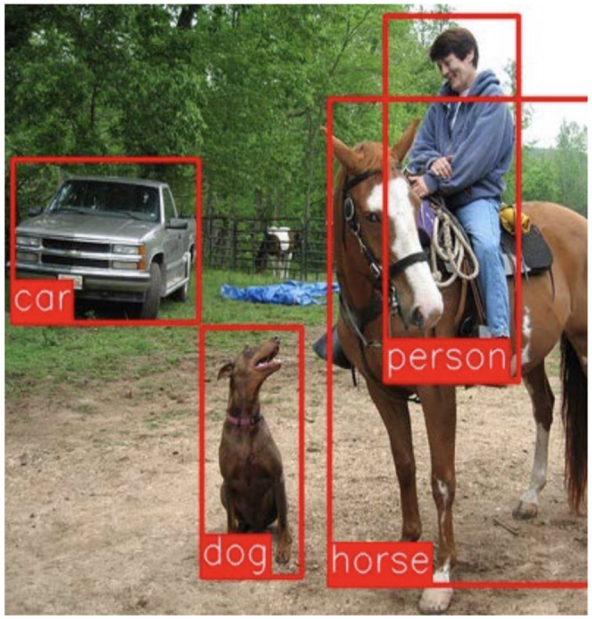
Beyond ML : From ANNs to Deep Neural Nets



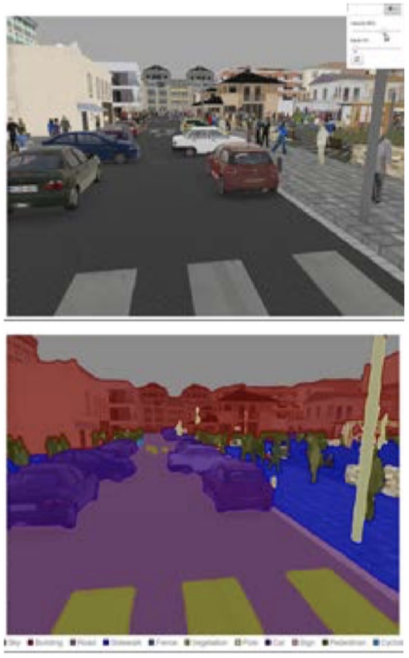
Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

Beyond Machine Learning : Deep Learning

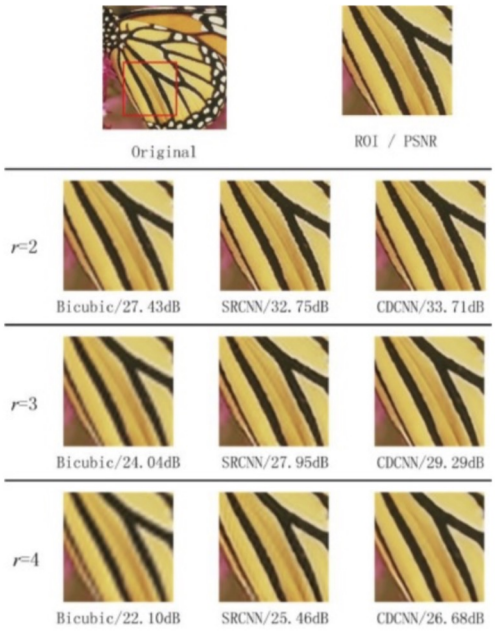
Combining machine learning and deep learning with huge datasets through convolutional neural networks (CNNs) to achieve powerful results.



Detection

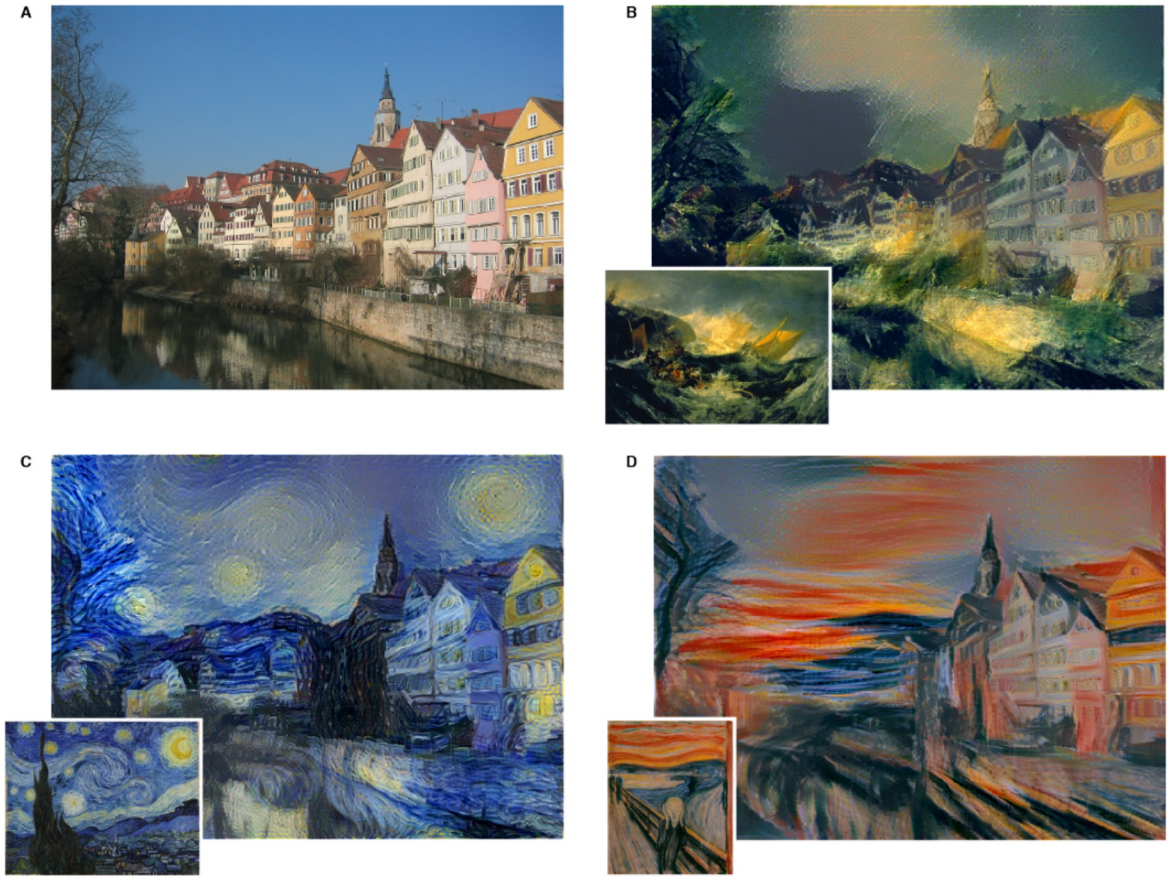


Segmentation



Denoising

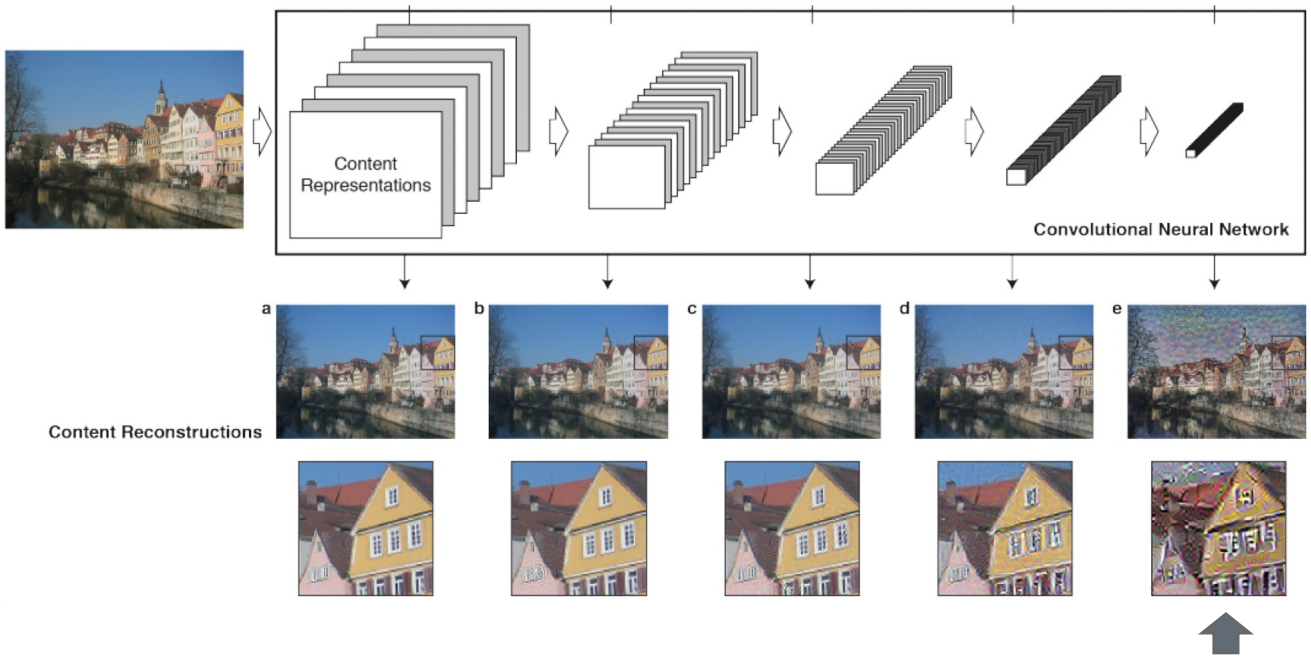
Beyond Machine Learning : Deep Learning



Understanding the structure of the neural networks one can distinguish textures from the global semantic of the image, and apply styles to images

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Beyond Machine Learning : Deep Learning



The convolutional NN tends to abstract more and more. Image reconstruction along the layers tend to disregard more and more the small details, keeping the general images.

Mixing this with lower layers that give the style of another image, can allow transferring styles.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

High level content preserved

Depth Cameras

Addition of 3D information can be achieved :

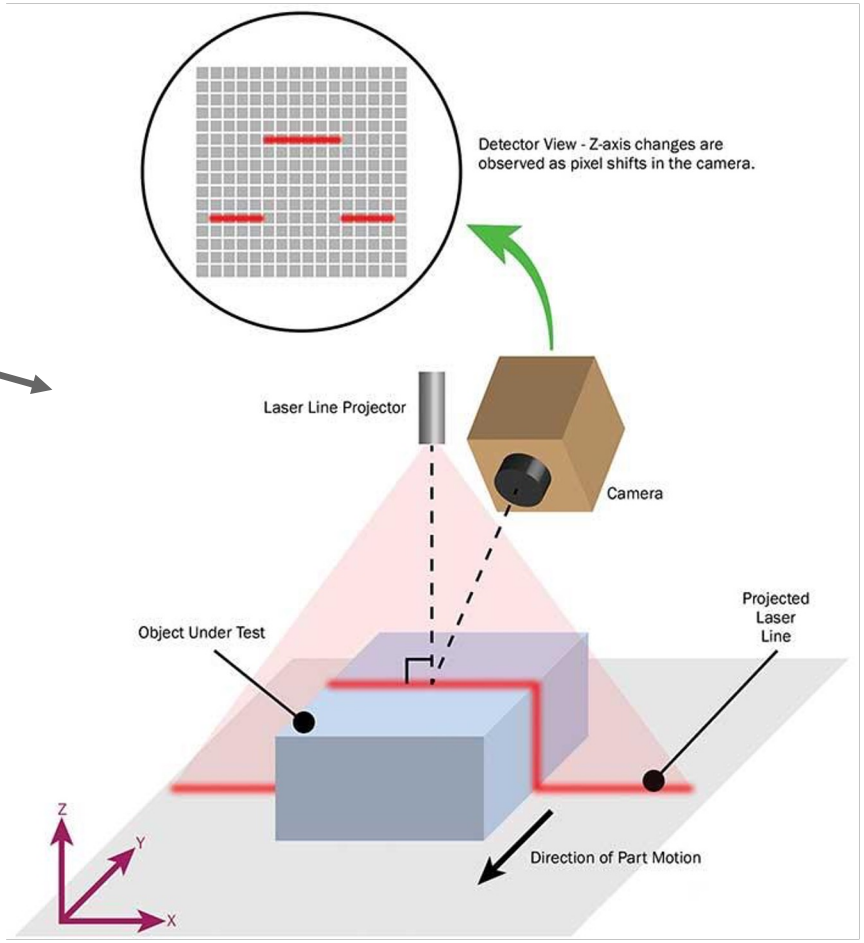
- by triangulation
- by stereovision
- by time of flight

https://maniaques.fr/test-mova-e30-avis/

Depth Cameras

Addition of 3D information can be achieved :

- by triangulation
- by stereovision
- by time of flight

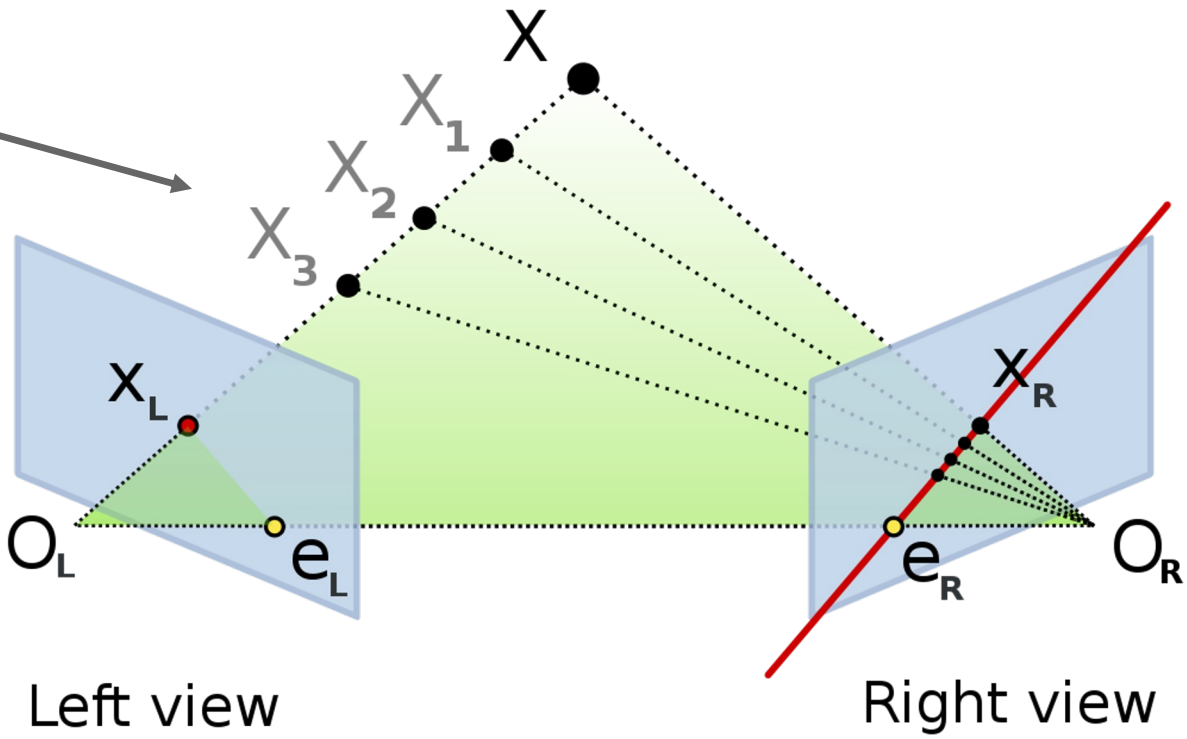


Courtesy of Coherent Inc.

Depth Cameras

Addition of 3D information can be achieved :

- by triangulation
- by stereovision
- by time of flight



Wikimedia, by Arne Nordmann

3D Reconstruction : Stereo Vision

Idealized camera geometry for stereo vision

- Disparity between two images -> computing of depth

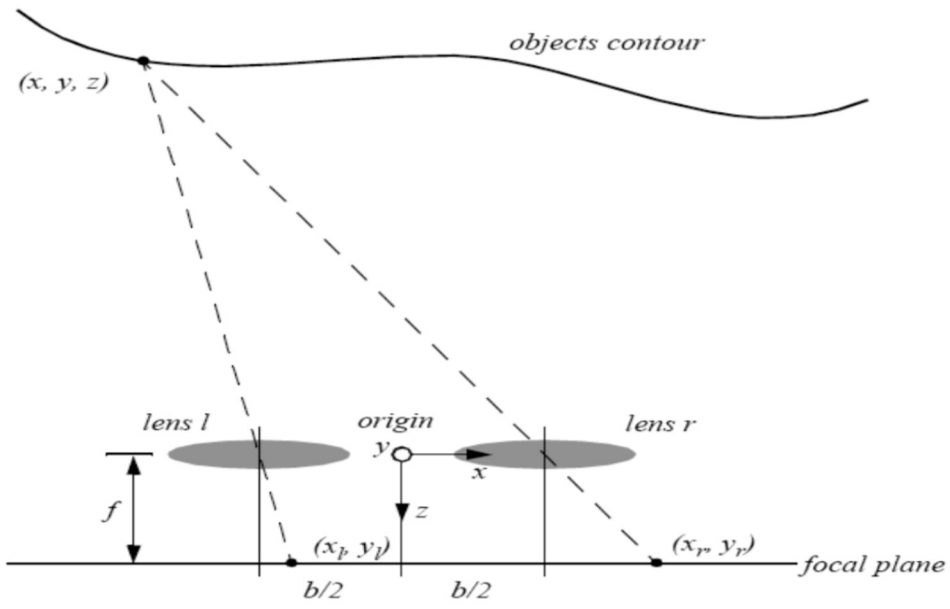
$$\frac{x_l}{f} = \frac{x + b/2}{z} \text{ and } \frac{x_r}{f} = \frac{x - b/2}{z}$$

disparity

$$\frac{x_l - x_r}{f} = \frac{b}{z}$$

$$z = b\frac{f}{x_l - x_r}$$

$$x = b\frac{(x_l + x_r)/2}{x_l - x_r} \; ; \; y = b\frac{(y_l + y_r)/2}{x_l - x_r}$$



3D Reconstruction : Stereo Vision

Distance is inversely proportional to disparity

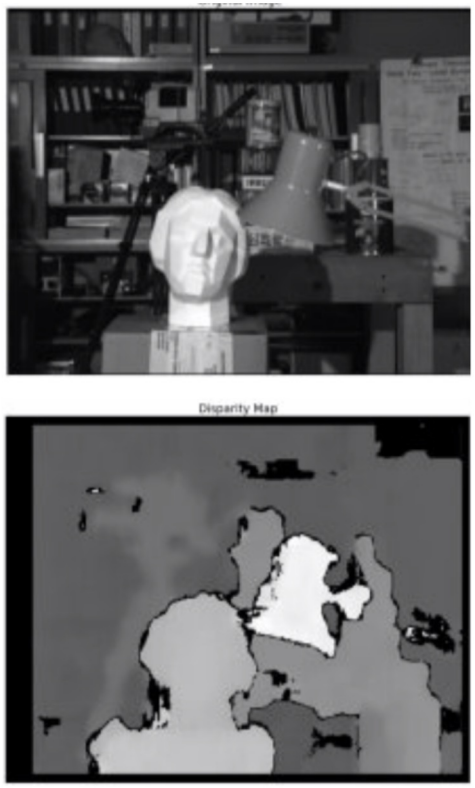
- Closer objects can be measured more accurately (nonlinear resolution).

Disparity is proportional to the baseline b

- For a given disparity error, the accuracy of the depth estimate increases with increasing baseline b.
- However, as b increases, some objects may appear in one camera, but not in the other (occlusion).

A point visible from both cameras is a *conjugate pair*

- In the standard arrangement of cameras, conjugate pairs lie on *epipolar lines* (parallel to the x-axis for the arrangement in the figure of the previous slide).



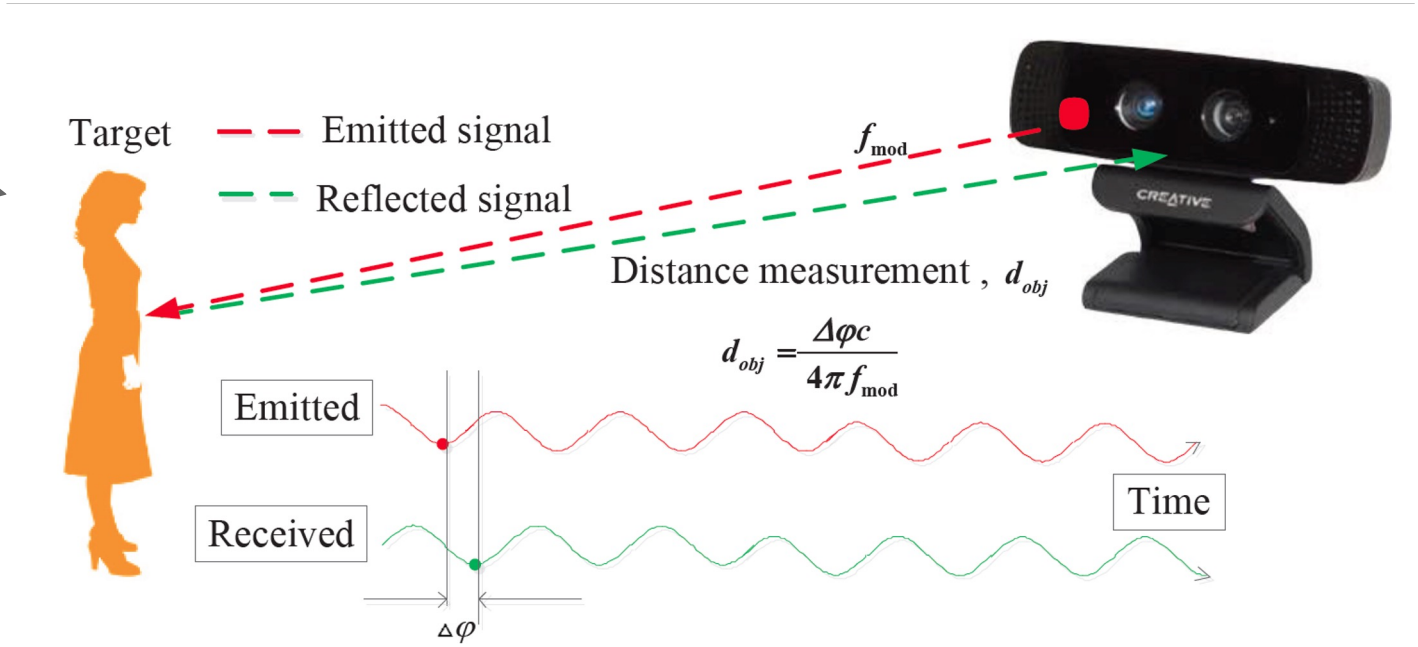
OpenCV 3.0 Depth map from Stereo

This can be extended to multiple cameras (multiple view vision). Similarly, camera poses are known. This still needs simple triangulation, with the only difference that the equations are a bit less pretty.

Depth Cameras

Addition of 3D information can be achieved :

- by triangulation
- by stereovision
- by time of flight (TOF)



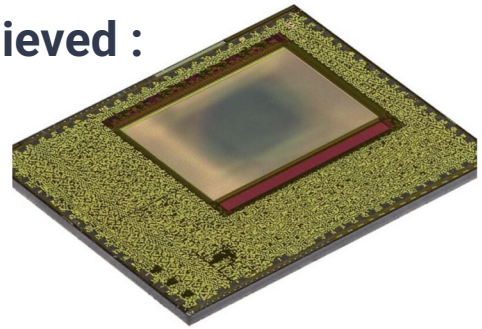
Islam, A., Hossain, M. A., & Jang, Y. M. (2016, July). Interference mitigation technique for time-of-flight (ToF) camera. In *2016 eighth international conference on ubiquitous and future networks (ICUFN)* (pp. 134-139). IEEE.

Depth Cameras

Addition of 3D information can be achieved :

- by triangulation
- by stereovision
- by time of flight (TOF)

160x240 = 38400



3D ToF imager enables smart navigation in vacuum cleaning robot

Business news | October 30, 2022

By Jean-Pierre Joosting

SUPPRESSION OF BACKGROUND ILLUMINATION

TIME OF FLIGHT | IMAGE SENSOR

NAVIGATION | ROBOTICS

Dreame, a leading Chinese based company specialized in smart home cleaning devices, has launched its new vacuum cleaning robot, the Dreame Bot W10 Pro. The smart vacuum cleaner is equipped with a camera featuring the REAL3™ Time of Flight (ToF) image sensor from Infineon Technologies AG and pmdtechnologies. The ToF imager enables smart navigation, 3D mapping, and excellent obstacle avoidance for service robots, cleaning robots and mops.

"The camera of Dreame's premium robot vacuum cleaner is equipped with our state-of-the-art 3D ToF imager technology. Combined with the latest technology for better contextual awareness from Dreame, the imager enables the cleaning device to identify and avoid obstacles," said Christian Herzum, Vice President Product-Line 3D-Sensing from Infineon. "Our advanced 3D vision engineering helps service robots learn from their daily cleaning routine and recognize household

The 3D ToF imager provides additional distance information for each of the cameras' 38000 pixels. It offers significantly improved 3D accuracy and precision compared to conventional

Depth Cameras: comparison

Looking at the features of sensors we looked at in the first lesson:

- Active VS passive: triangulation and TOF are active. Stereovision is passive, which is an advantage in some situations, ensuring no cross-sensitivity.
- Because active, triangulation and TOF are less sensitive, for instance, to the visual texture. Stereovision is based on the visual texture of the environment and without texture cannot work.
- Because active, triangulation and TOF can be impacted if there is something (sun light for instance) that strongly interfere with their emitted signal. They also rely on the reflection of the emitted signal (too absorbing surfaces could be an issue)
- Because triangulation and stereovision are based on geometrical properties, their resolution depends on the distance, and have less resolution in the long distance. TOF has in general a constant resolution in respect to distance.

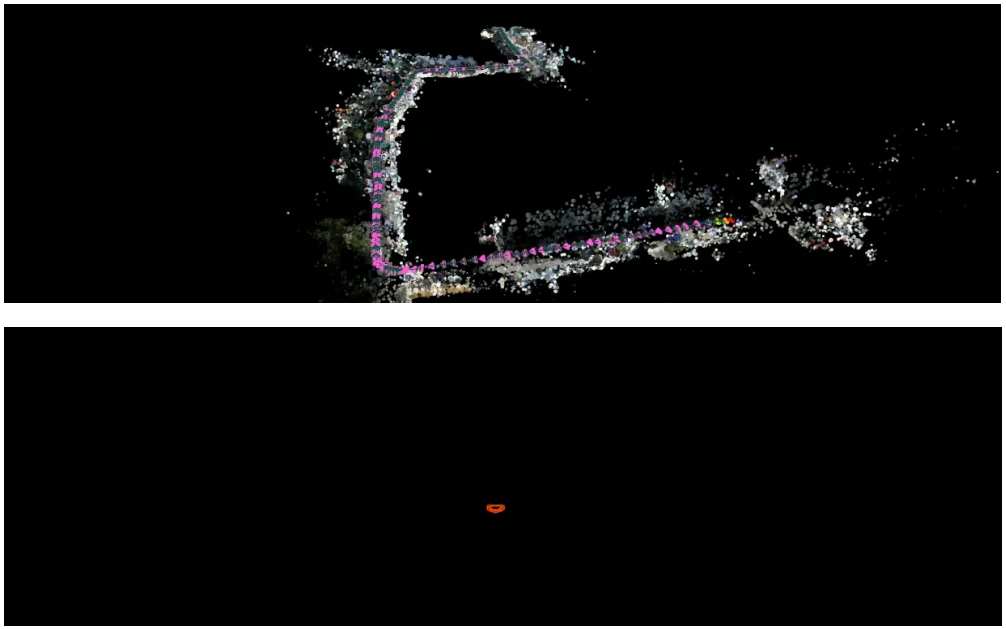
3D Reconstruction : SLAM

Simultaneous localization and mapping:

• Before either had : (known map and unknown camera poses) or (known camera poses and unknown map)

• Now : unknown map and unknown camera poses.

• Solution requires non-linear optimization.



What should I remember

- Machine learning
  - Definition
  - Statistical approach
    - Principle of the linear discriminant analysis (LDA)
    - when LDA works and when it does not work
  - Artificial neural networks (ANN) approach
    - Definition of an artificial neuron
    - ANN structures: when linear separator and when not
    - ANN with memory: concept
    - Types of learning methods : definitions
    - Deep neural networks
      - concept
      - application to image analysis (black box)
- Depth cameras
  - Three approaches: principle, advantages and disadvantages