Robotics

Miao Li

Fall 2023, Wuhan University

WeChat: 15527576906

Email: limiao712@gmail.com

Goal for this course

- **Design：soft hand design x1**
- **Perception: vision, point cloud, tactile, force/torque x1**
- **Planning: sampling-based, optimization-based, learning-based x3**
- **Control: feedback, multi-modal x2**
- **Learning: imitation learning, RL x2**
- **Simulation tool (pybullet, matlab, OpenRAVE, Issac Nvidia, Gazebo)**
- **How to get a robot moving!**

Today’s Agenda

- **What is robot perception? (~12)**
- **Robot vision and computer vision (~5)**
- **Force sensing (~5)**
- **Tactile sensing (~5)**
- **Challenges of robot perception (10)**
- **Algorithms for perception**
  - **State estimation (~5)**
  - **End to end learning (~5)**
  - **Active perception (~5)**
- **Quick Review of Deep Learning (~20)**

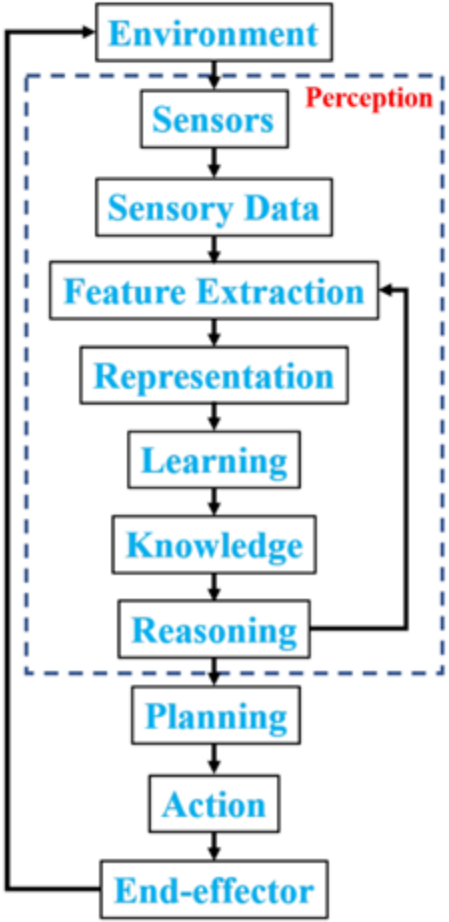
Incredible human skills

Incredible animal skills

Robotics – Learn the mapping from perception to action

Tesla's Optimus Robot Sort Objects Autonomously
https://www.youtube.com/watch?v=oL5YNtDUQXU&ab_channel=CNETHighlights

Robotics – Learn the mapping from perception to action



Robotics – Learn the mapping from perception to action

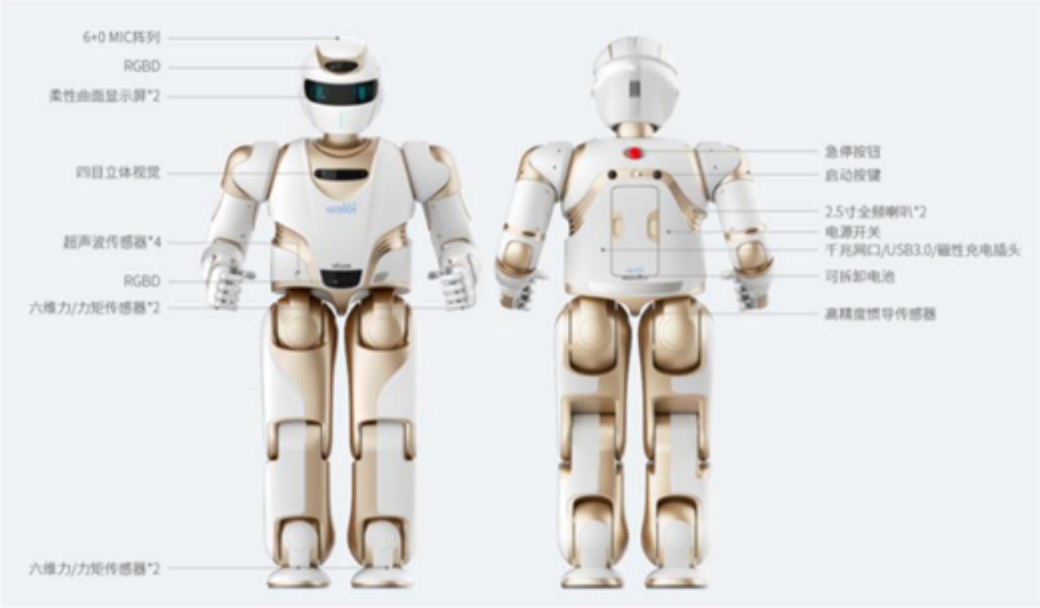
Why robot perception?

Making sense of the unstructured environment …

- Incomplete knowledge of the scene
- Imperfect actions may lead to failure
- Environment dynamics

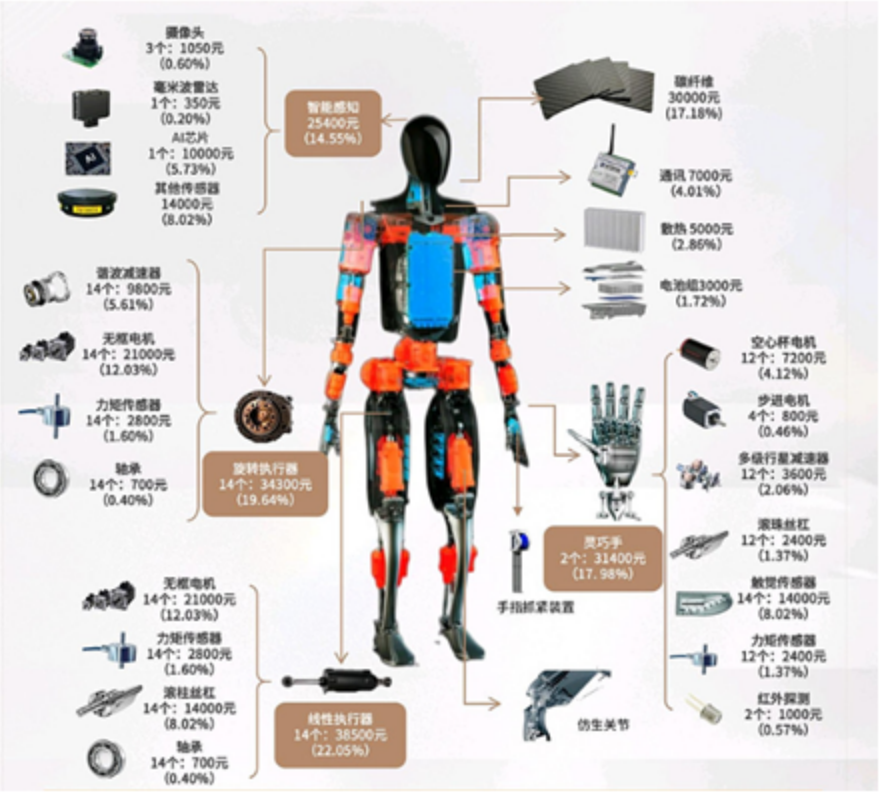
Robot sensor

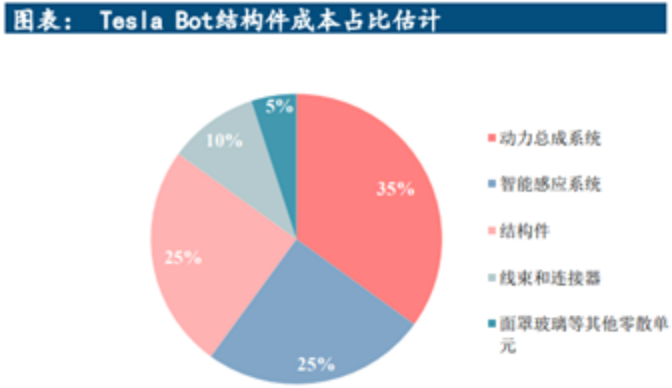
Understand the real world through different sensors

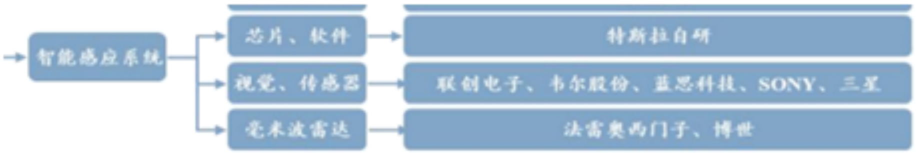


Robot sensor

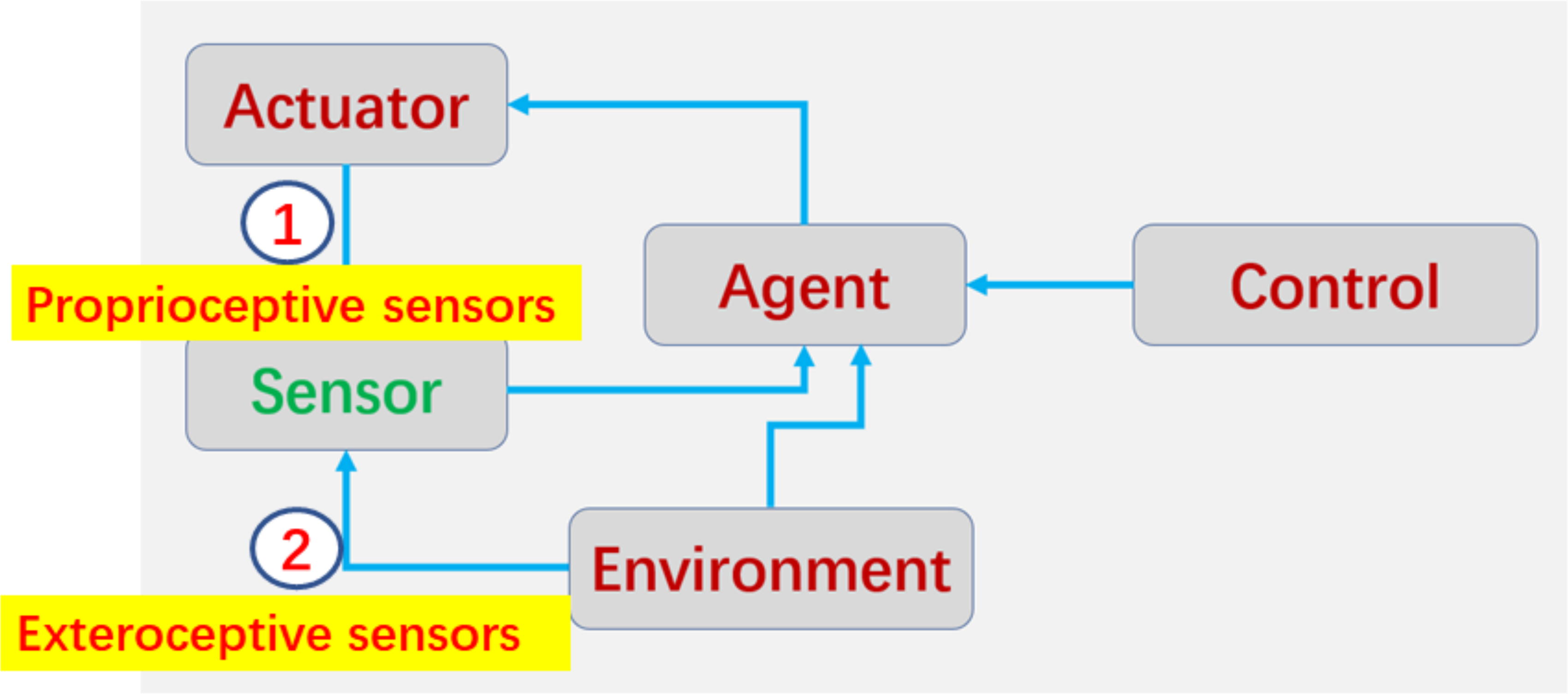
Understand the real world through different sensors







Robot sensor



Robot sensor

- proprioceptive sensors measure the internal state of the robot (position and velocity of joints, but also torque at joints or acceleration of links)
  - kinematic calibration, identification of dynamic parameters, control
- exteroceptive sensors measure/characterize robot interaction with the environment, enhancing its autonomy(forces/torques, proximity, **vision**, but also sensors for sound, smoke humidity, …)
  - control of interaction with the environment, obstacle avoidance localization of mobile robots, navigation in unknown environments

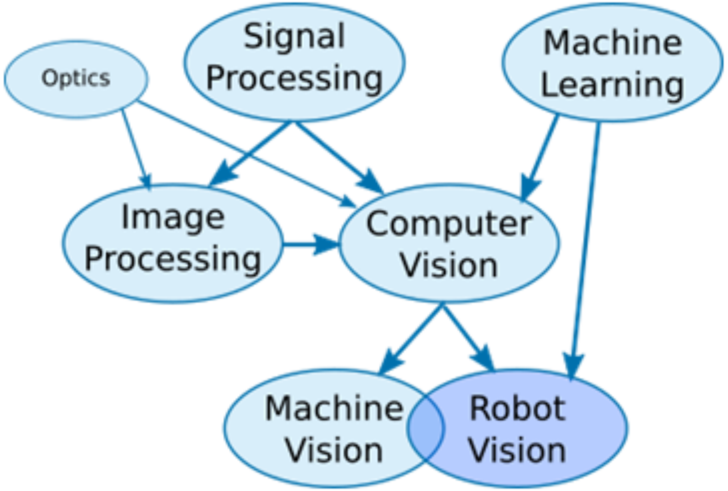
Robot vision vs. Computer vision

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g. in the forms of decision.

Deep Learning

Robot vision vs. Computer vision

Technique	Input	Output
Signal Processing	Electrical signals	Electrical signals
Image Processing	Images	Images
Computer Vision	Images	Information/features
Pattern Recognition/Machine Learning	Information/features	Information
Machine Vision	Images	Information
Robot Vision	Images	Physical Action



https://blog.robotiq.com/robot-vision-vs-computer-vision-whats-the-difference

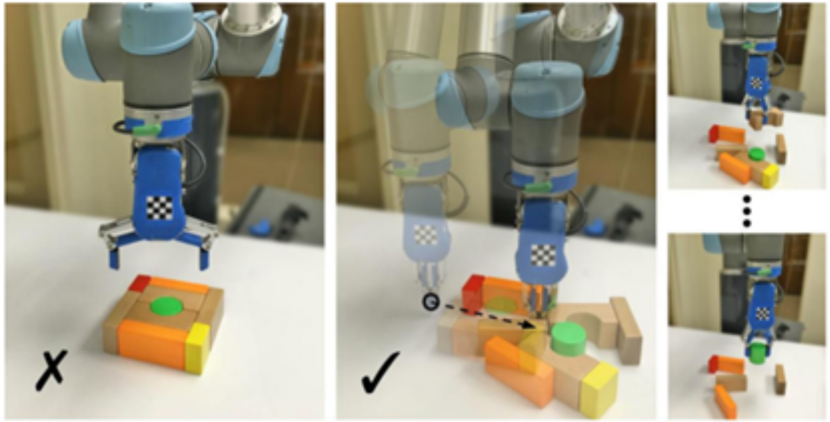
Robot vision vs. Computer vision

- Robot vision is **embodied**, **active**, and environmentally **situated**.

- **Embodied**: Robots have physical bodies and experience the world directly. Their actions are part of a dynamic with the world and have immediate feedback on their own sensation.
- **Active**: Robots are active perceivers. It knows why it wishes to sense, and chooses what to perceive, and determines how, when and where to achieve that perception.
- **Situated**: Robots are situated in the world. They do not deal with abstract descriptions, but with the here and now of the world directly influencing the behaviorof the system.

Robot vision vs. Computer vision

[Levine et al., IJRR 2016]



[Zeng et al., IROS 2018]

2D camera

1963 – Lawrence Roberts, the Father of Computer Vision publishes "Machine Perception Of Three-Dimensional Solids" where he discusses extracting 3D information about solid objects from 2D images. This lead to much research in MIT's artificial intelligence lab and other research institutions looking at computer vision in the context of blocks and simple objects.

1966 – The summer project at MIT marks the landmark in the development of pattern recognition.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

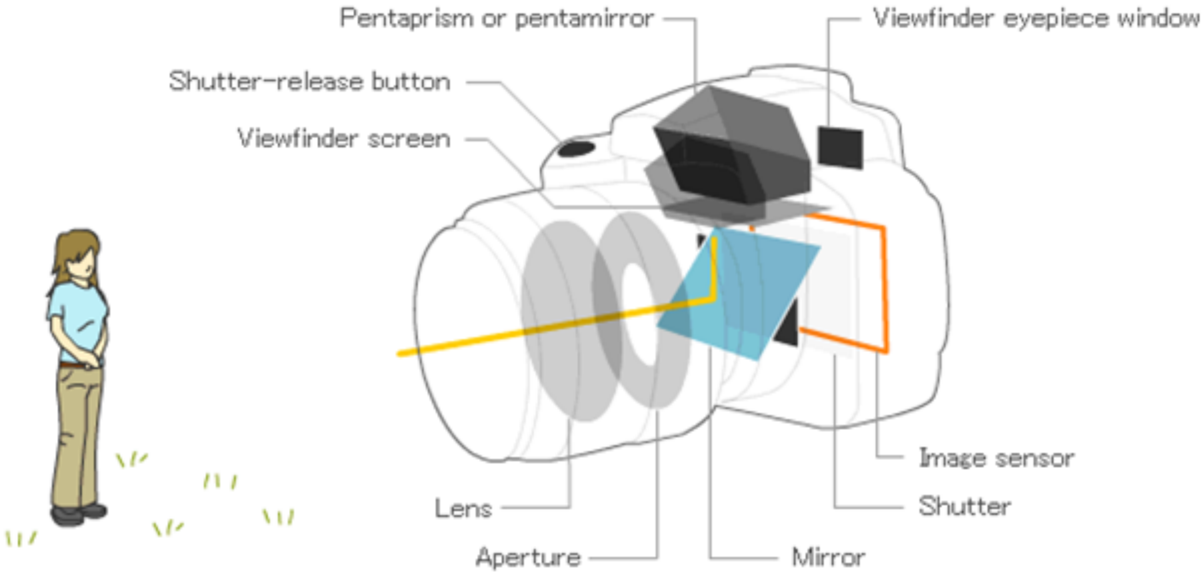
THE SUMMER VISION PROJECT

Seymour Papert

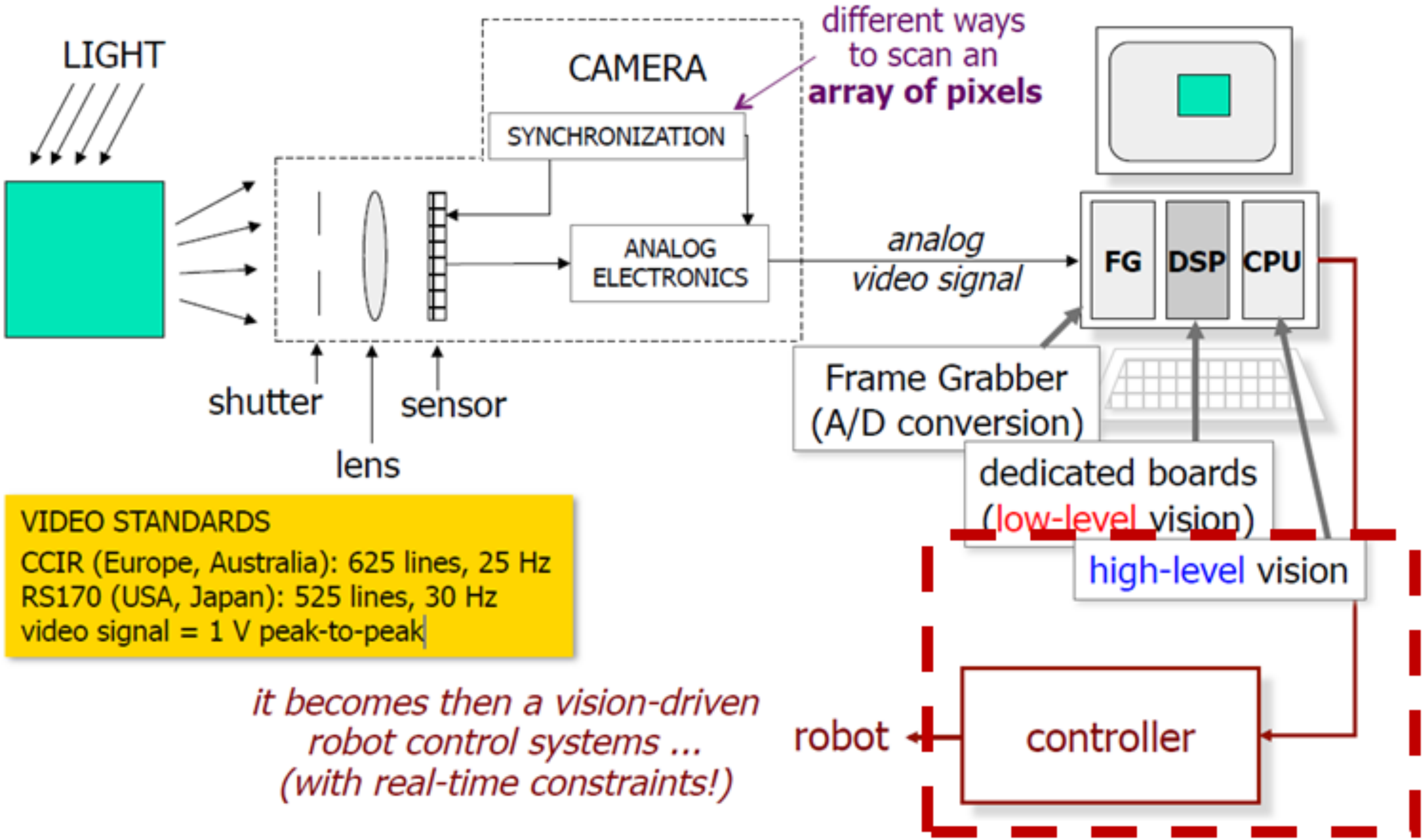
https://emergentvisiontec.com/tech-portal/evolution-of-machine-vision/

2D camera

The Optical Path from the Lens Through the Mirror to the Viewfinder



2D camera



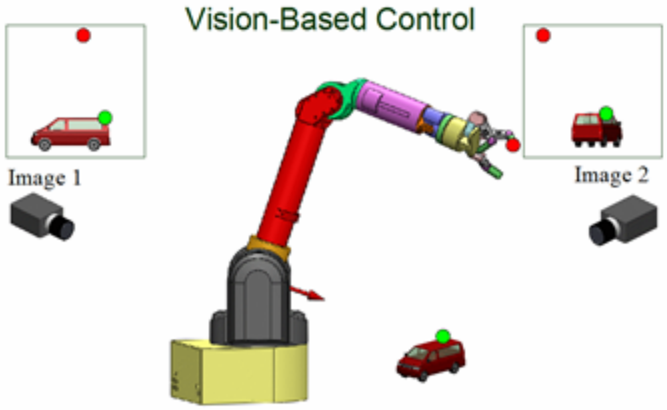
LIGHT
CAMERA
SYNCHRONIZATION
ANALOG ELECTRONICS
different ways to scan an **array of pixels**
analog video signal
FG
DSP
CPU
shutter
lens
sensor
Frame Grabber (A/D conversion)
dedicated boards (low-level vision)
high-level vision
VIDEO STANDARDS
CCIR (Europe, Australia): 625 lines, 25 Hz
RS170 (USA, Japan): 525 lines, 30 Hz
video signal = 1 V peak-to-peak
it becomes then a vision-driven robot control systems ... (with real-time constraints!)
robot
controller

2D camera

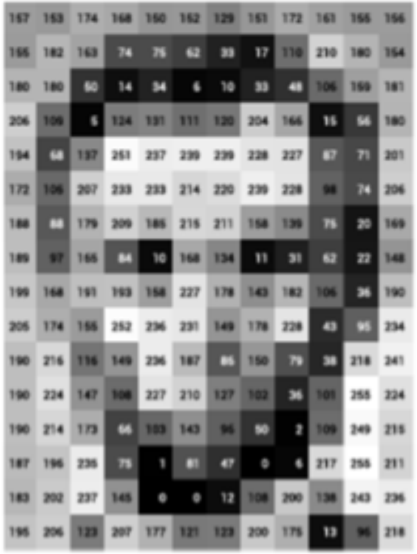
Visual servoing, also known as **vision-based robot control** and abbreviated **VS**, is a technique which uses feedback information extracted from a vision sensor (visual feedback) to control the motion of a robot. One of the earliest papers that talks about visual servoing was from the SRI International Labs in 1979.

1. *"Basic Concept and Technical Terms". Ishikawa Watanabe Laboratory, University of Tokyo. Retrieved 12 February 2015.*

2.^ Agin, G.J., "Real Time Control of a Robot with a Mobile Camera". Technical Note 179, SRI International, Feb. 1979.



2D image format



What the computer sees

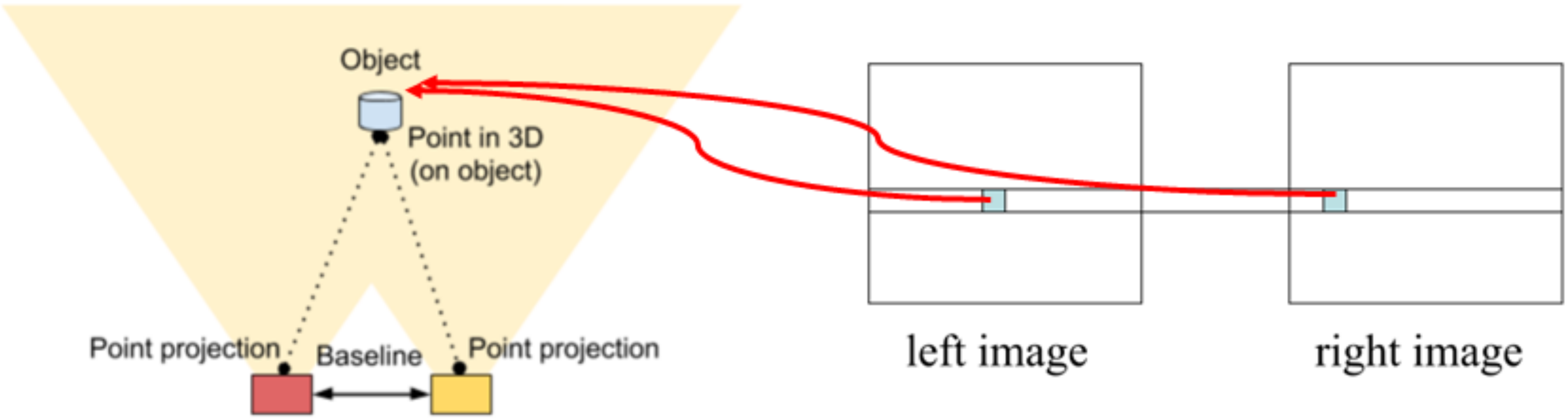
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

Slide Credit: Ava Soleimany, MIT

Images Are Numbers

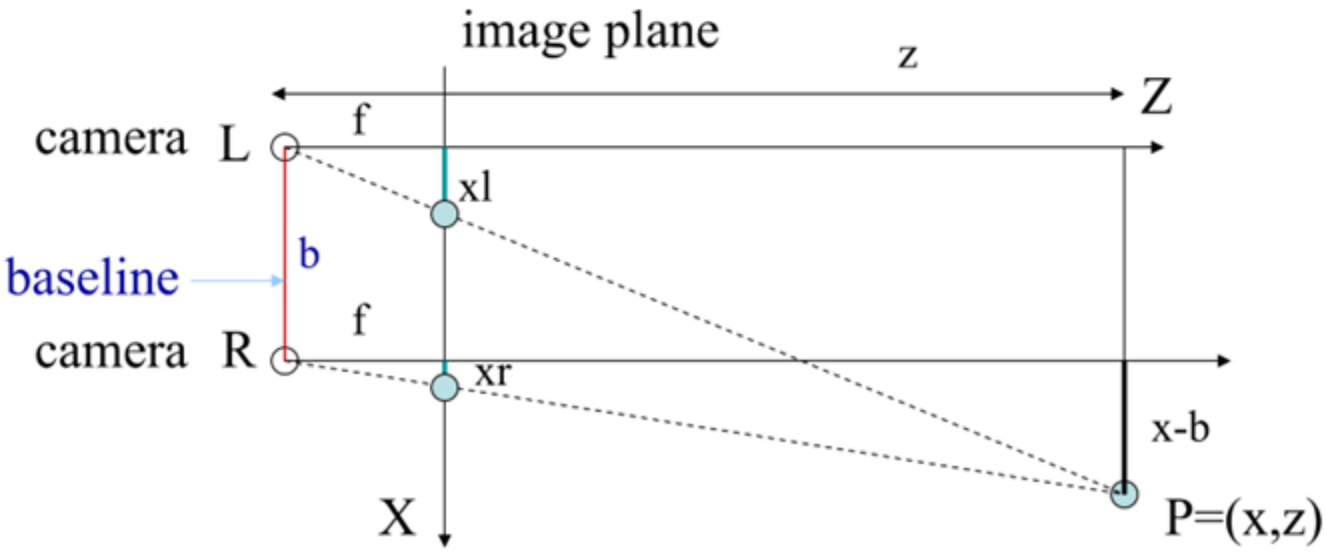
3D from stereo



disparity: the difference in image location of the same 3D point when projected under perspective to two different cameras.

$$\boxed{d = xleft - xright}$$

3D from stereo



$$\frac{z}{f} = \frac{x}{xl}$$

$$\frac{z}{f} = \frac{x-b}{xr}$$

$$\frac{z}{f} = \frac{y}{yl} = \frac{y}{yr}$$

y-axis is perpendicular to the page.

3D from stereo

For stereo cameras with parallel optical axes, focal length f, baseline b, corresponding image points (xl,yl) and (xr,yr) with disparity d:

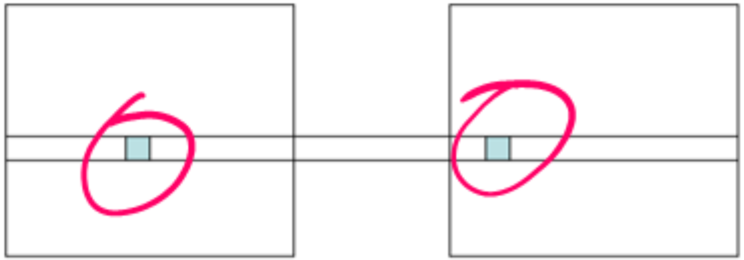
$$z = f*b / (xl - xr) = f*b/d$$

$$x = xl*z/f \quad \text{or} \quad b + xr*z/f$$

$$y = yl*z/f \quad \text{or} \quad yr*z/f$$

This method of determining depth from disparity is called **triangulation.**

3D from stereo



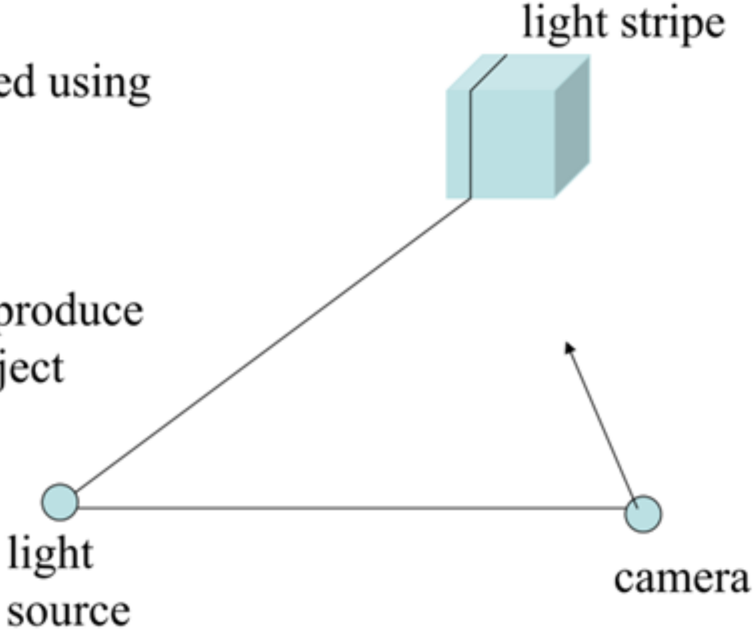
left image　　　　right image

We need to find the "correspondence".

3D from structure

3D data can also be derived using

- a single camera
- a light source that can produce stripe(s) on the 3D object

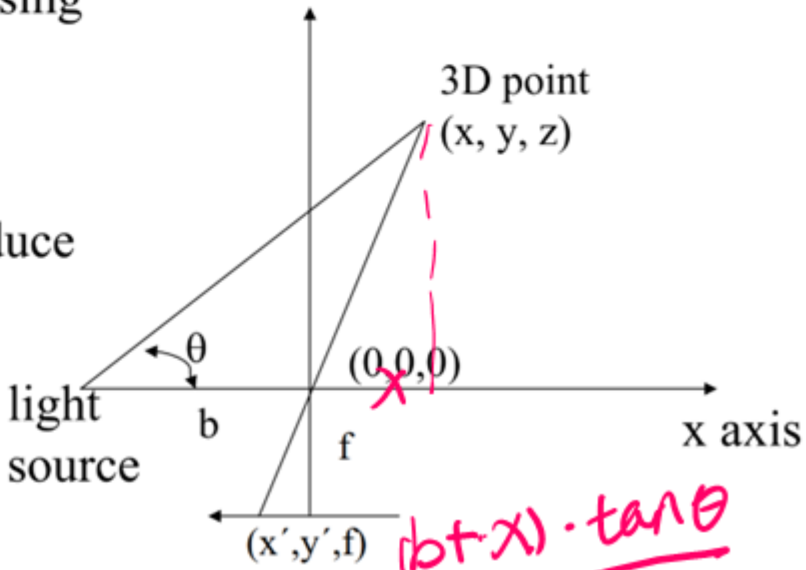


3D from structure

3D data can also be derived using

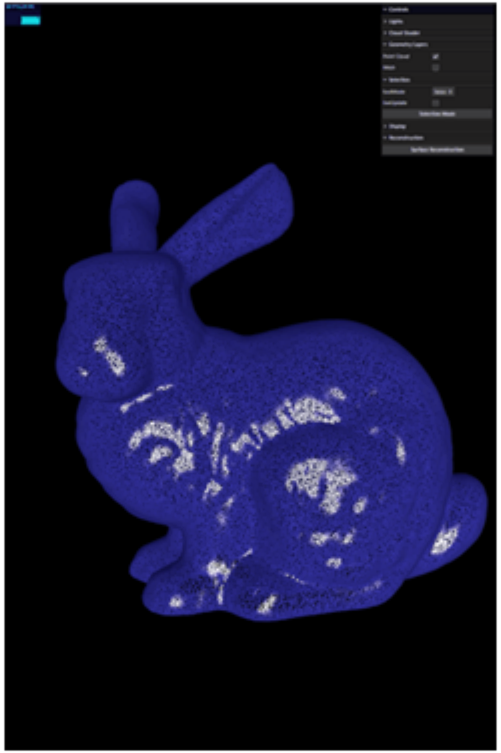
- a single camera
- a light source that can produce stripe(s) on the 3D object

$$\underbrace{[x\ y\ z]}_{\text{3D}} = \frac{b}{f\cot\theta - x'}\,\underbrace{[x'\ y'\ f]}_{\text{image}}$$



$$\frac{x}{x'} = \frac{(b+x)\cdot\tan\theta}{f}$$

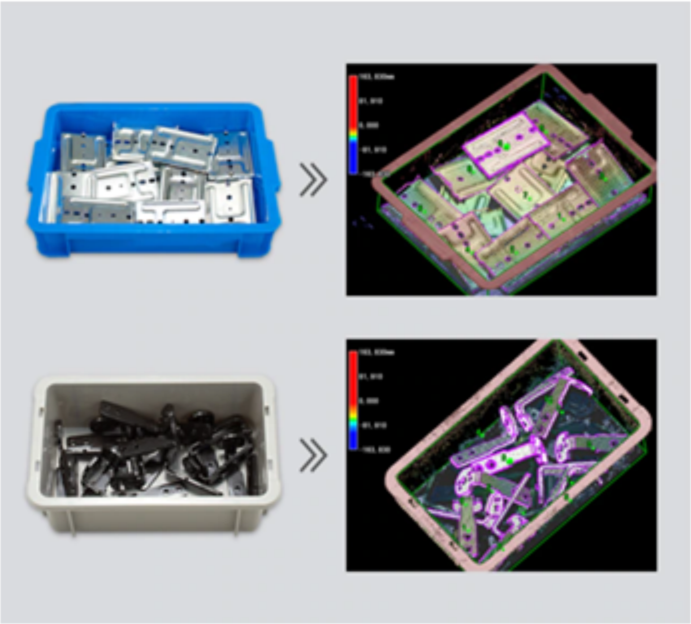
3D vision format



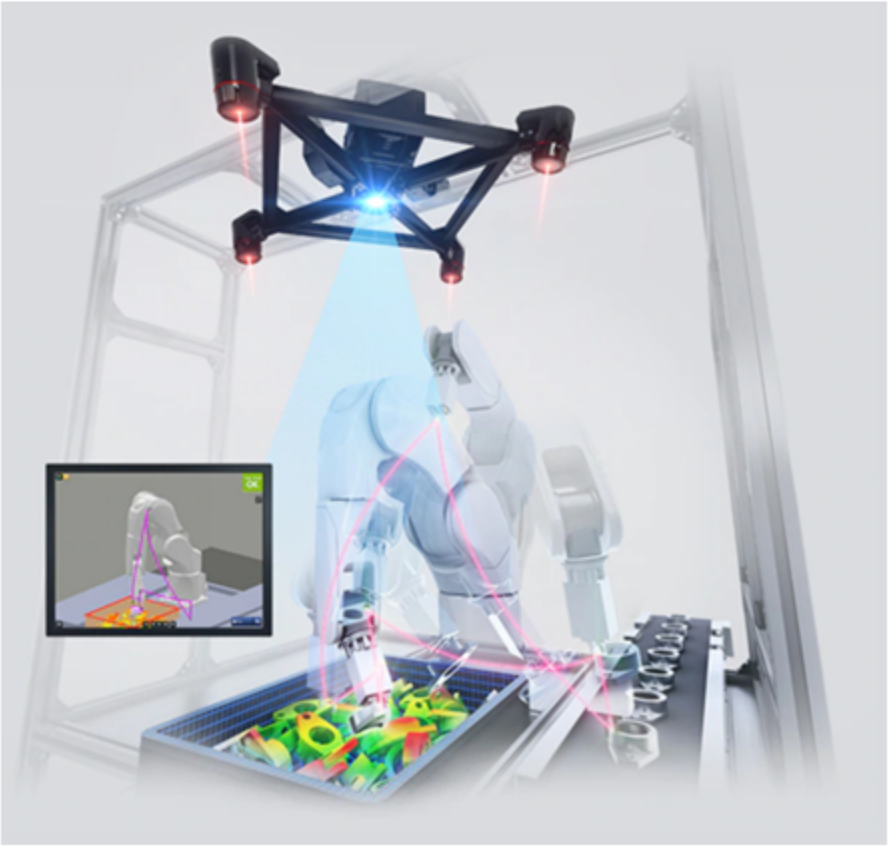
- PLY - a polygon file format, developed at Stanford University by Turk et al
- STL - a file format native to the stereolithography CAD software created by 3D Systems
- OBJ - a geometry definition file format first developed by Wavefront Technologies
- X3D - the ISO standard XML-based file format for representing 3D computer graphics data
- and many others

https://pointclouds.org/documentation/tutorials/pcd_file_format.html

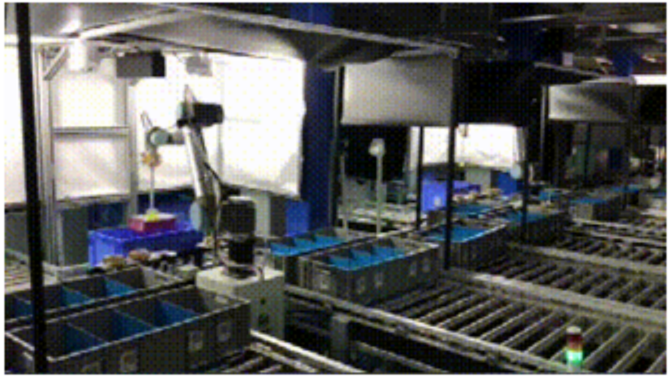
3D camera application

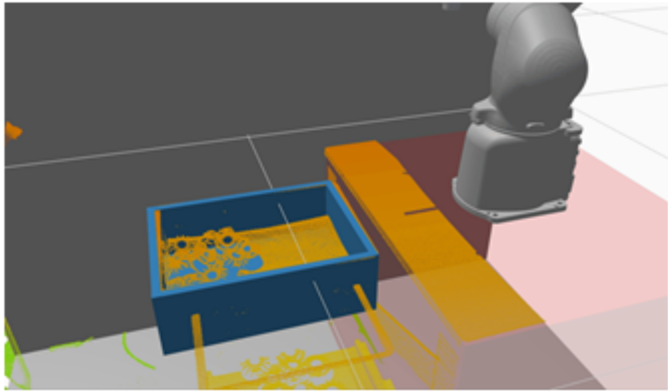


Source: Keyence website



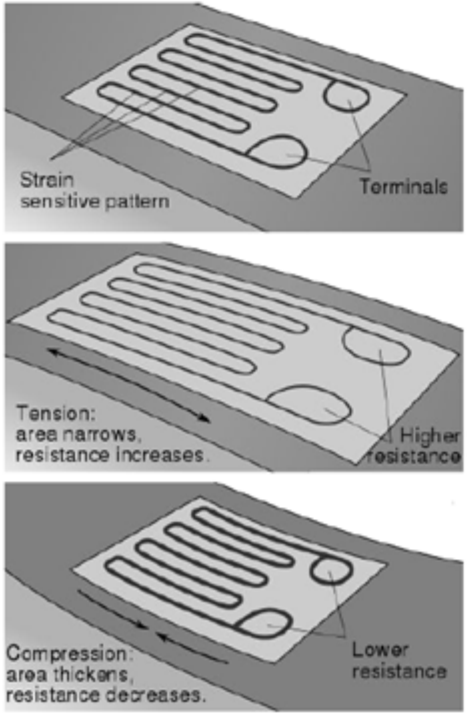
3D camera application





F/T sensor principle

- indirect information obtained from the measure of deformation of an elastic element subject to the force or torque to be measured
- basic component is a *strain gauge*: uses the variation of the resistance R of a metal conductor when its length L or cross-section S vary

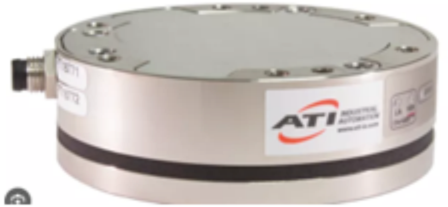


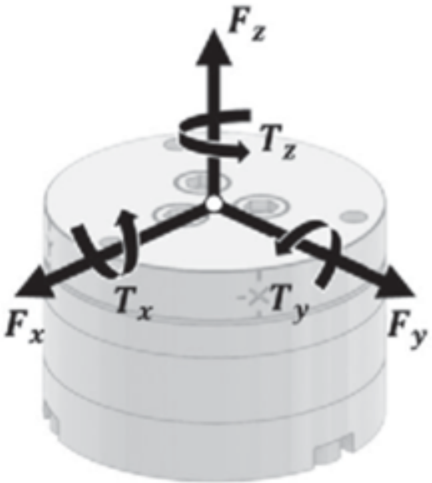
$$\frac{\partial R}{\partial L} > 0 \qquad \frac{\partial R}{\partial S} < 0$$

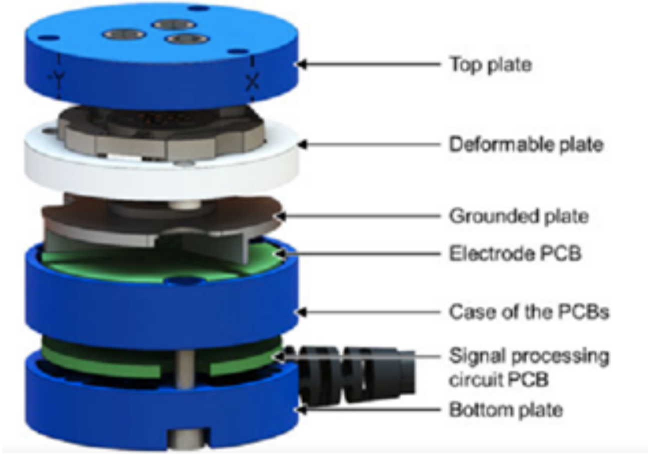
$$\frac{\partial R}{\partial T} \text{ small}$$

temperature

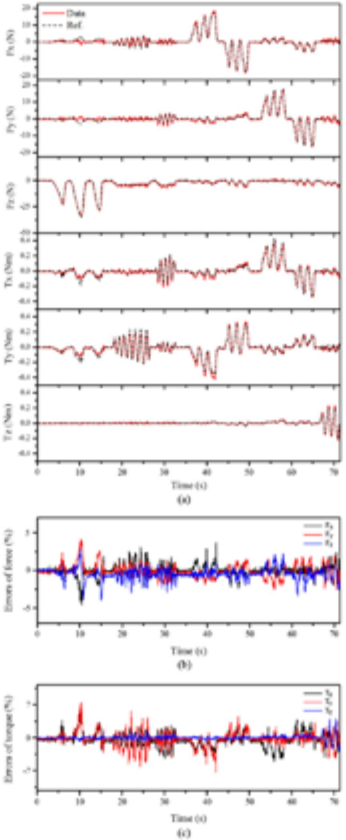
F/T sensor principle



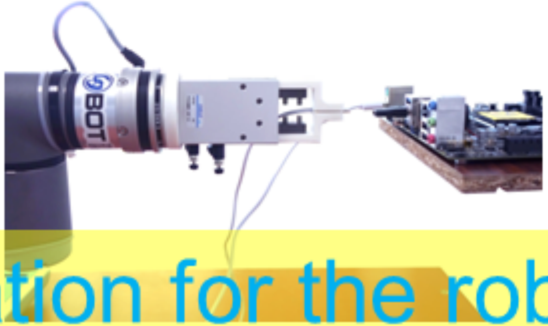




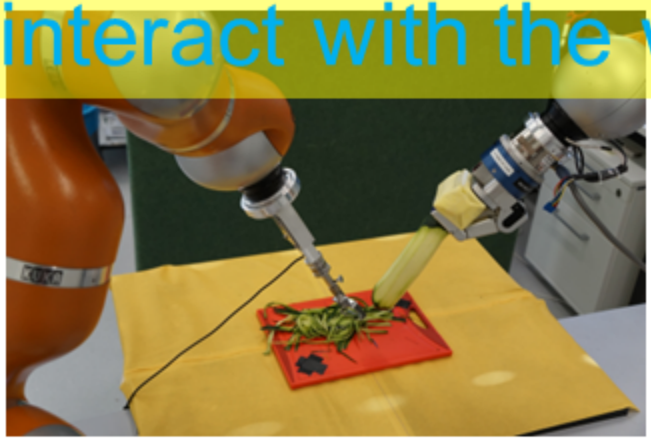
F/T information format

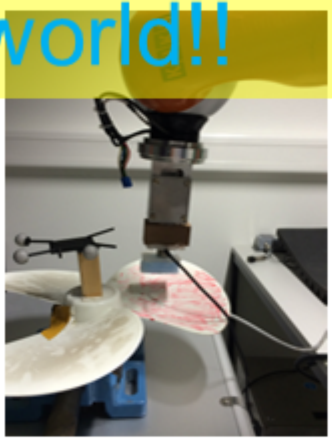


F/T sensor application

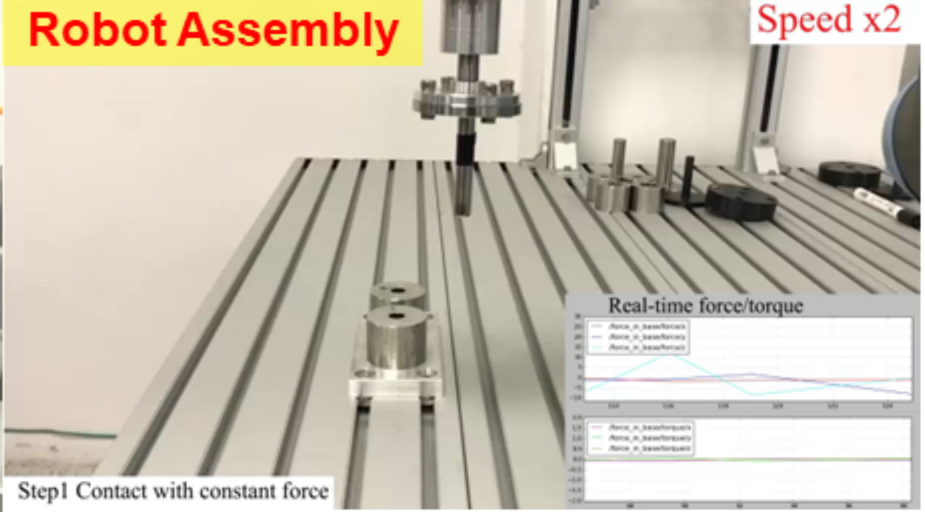


Force is an essential information for the robots to physically interact with the world!!



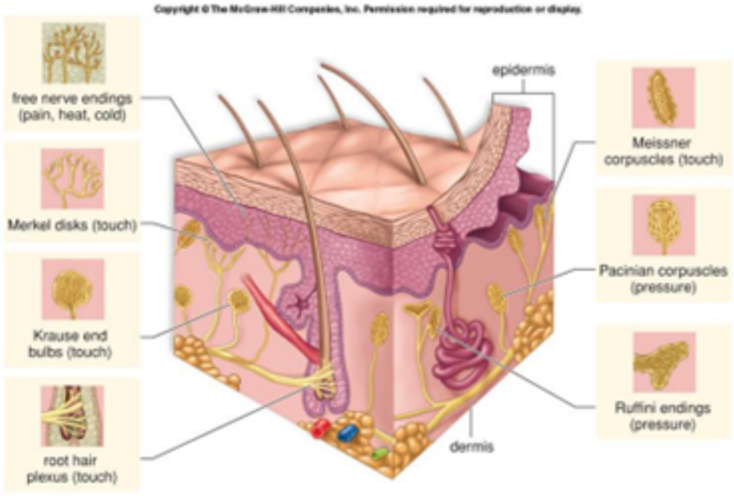


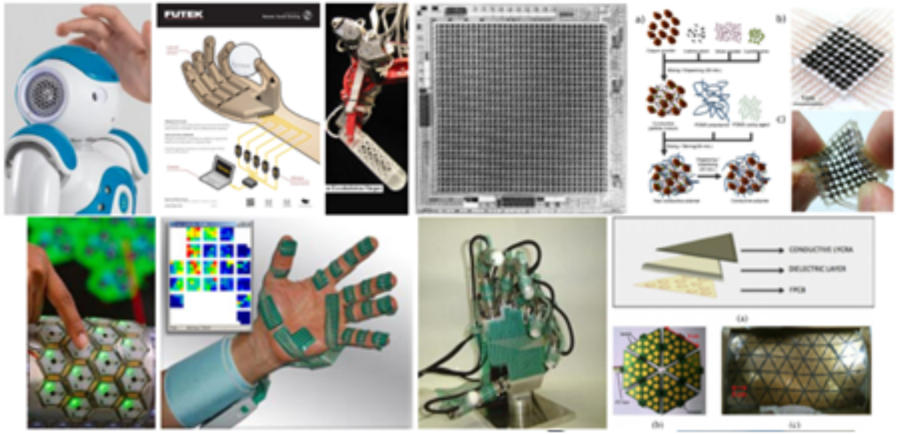

Robot Polishing


Robot Assembly
Speed x2
Real-time force/torque
Step1 Contact with constant force

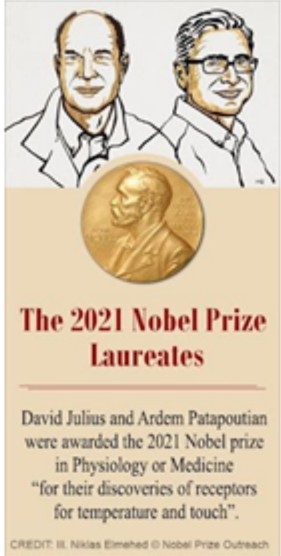

Kinesthetic Teaching

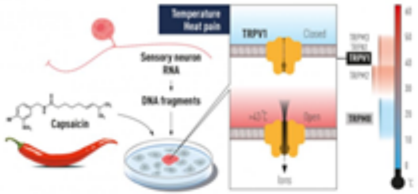
Tactile sensor principle

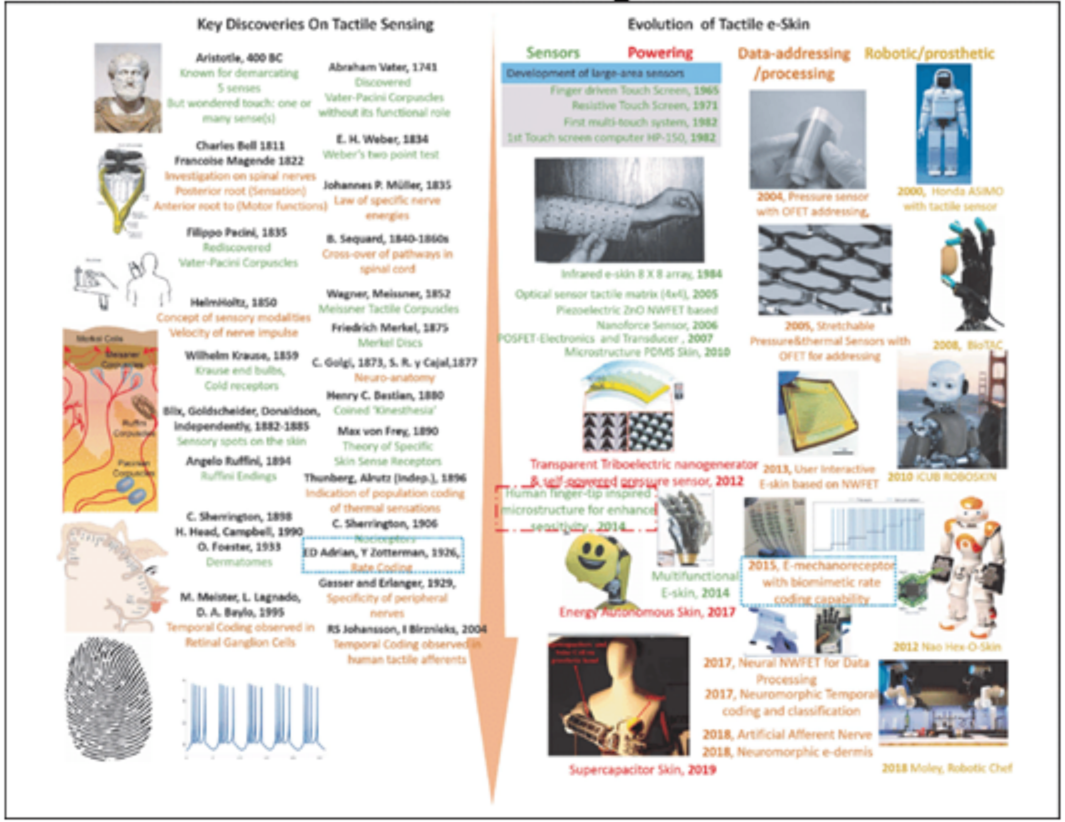




Tactile sensor history

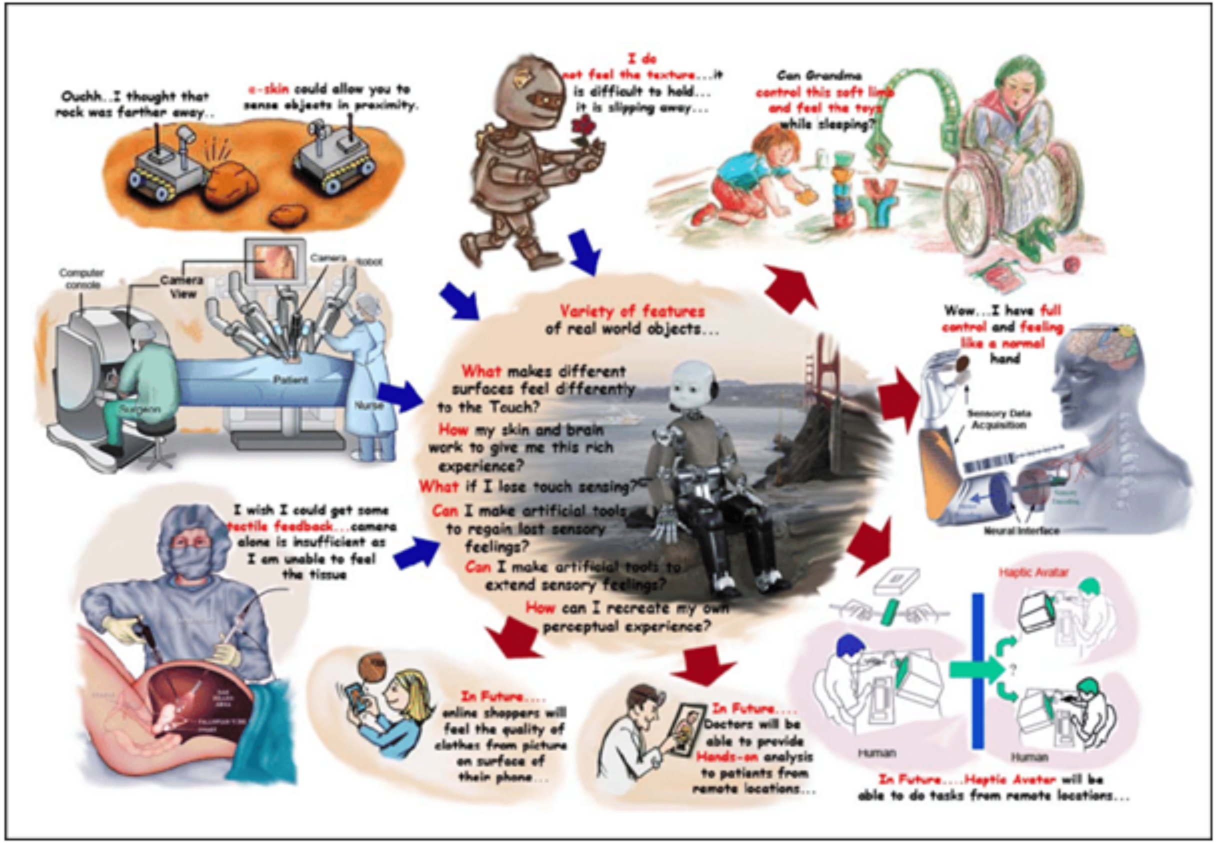




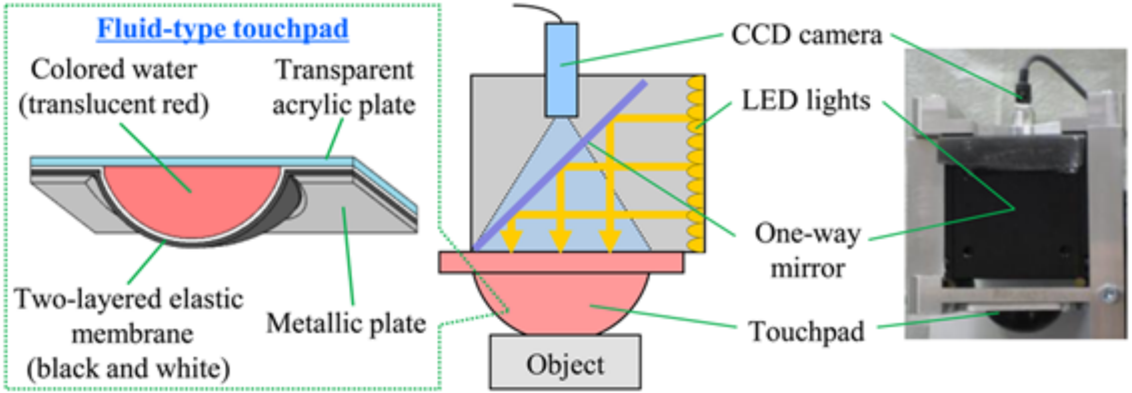


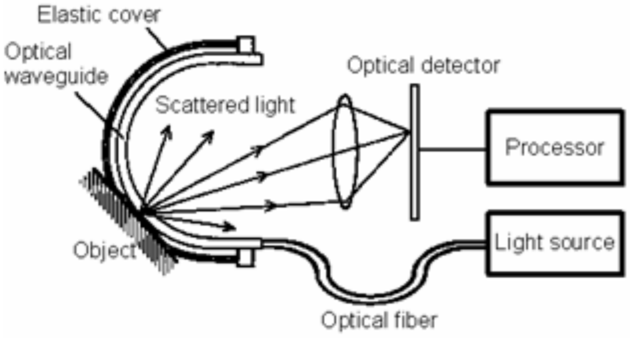
[Large-Area Soft e-Skin: The Challenges Beyond Sensor Designs]

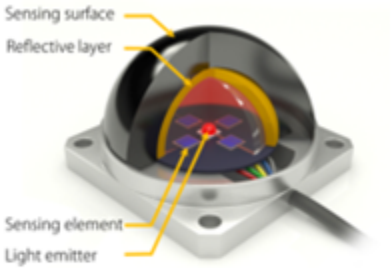
Tactile sensor applications



Tactile sensor principle







Robot Skin

Why is “optical skin” a good idea?

- Easy to get millions of sensors (pixels)
- Bonus: Proximity Sense.
- Separate deformation and its measurement.
- Outer layer has nothing in it. Cheap to repair or replace. Can be optimized for desired mechanical and lifetime properties.
- Connections (solder joints etc.) and wires aren’t deforming.
- Reliability – fewer wires, components. Cameras are already reliable.
- Whole-body vision: reduce occlusion.
- Avoid rigid printed circuit boards or chips.

Finger Vision: Proximity Sensing
Seeing with your fingers

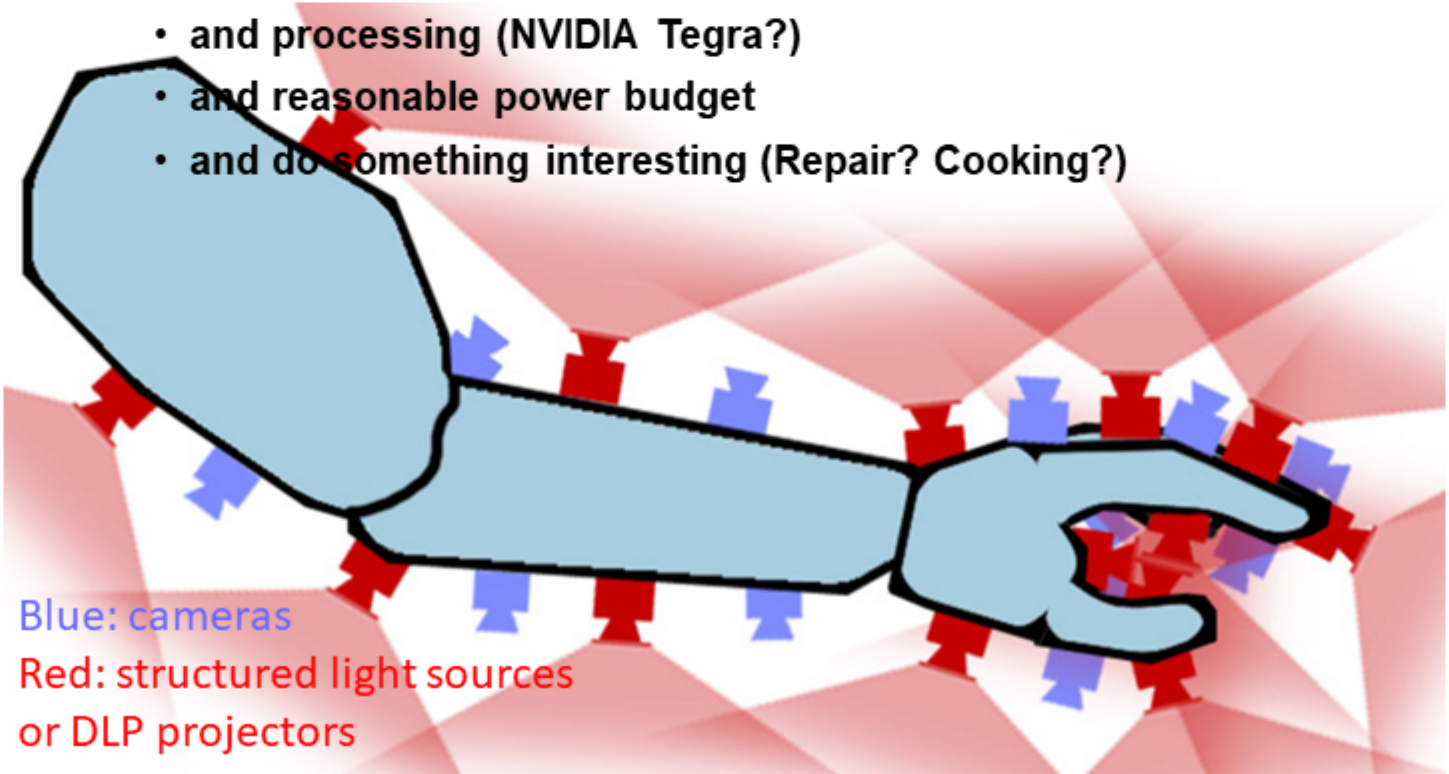
Whole Body Vision: Origins

Argus (Greece)

Hyakume (Japan)

Goal and Challenge

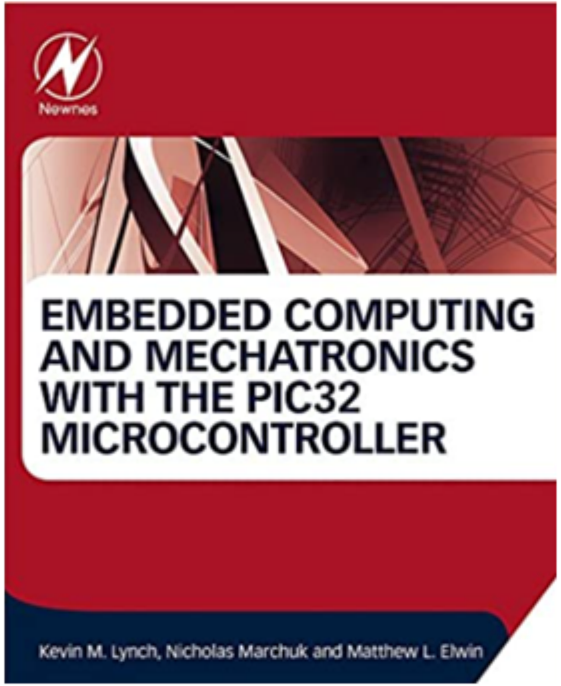
- Can we build 100 camera system on full robot
- with full multimodal sensor suite
- and networking (GigE?)
- and processing (NVIDIA Tegra?)
- and reasonable power budget
- and do something interesting (Repair? Cooking?)

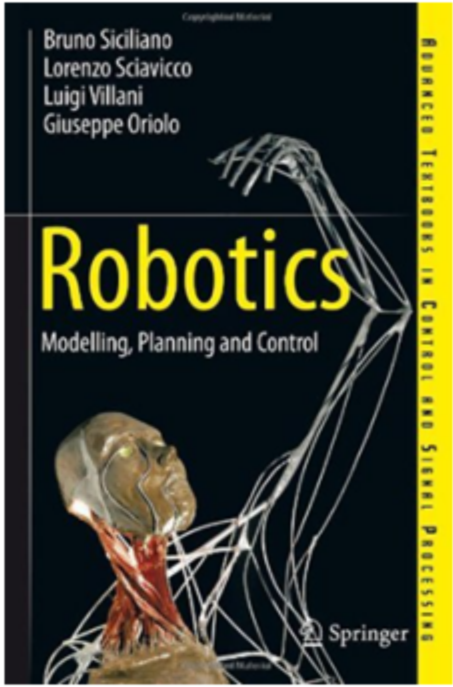


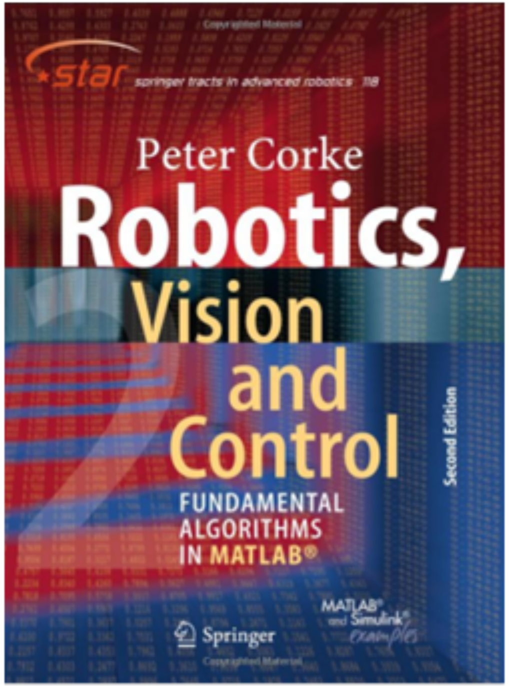
Blue: cameras

Red: structured light sources or DLP projectors

Some reference books on sensors using in mechatronics, robotics, vision control…







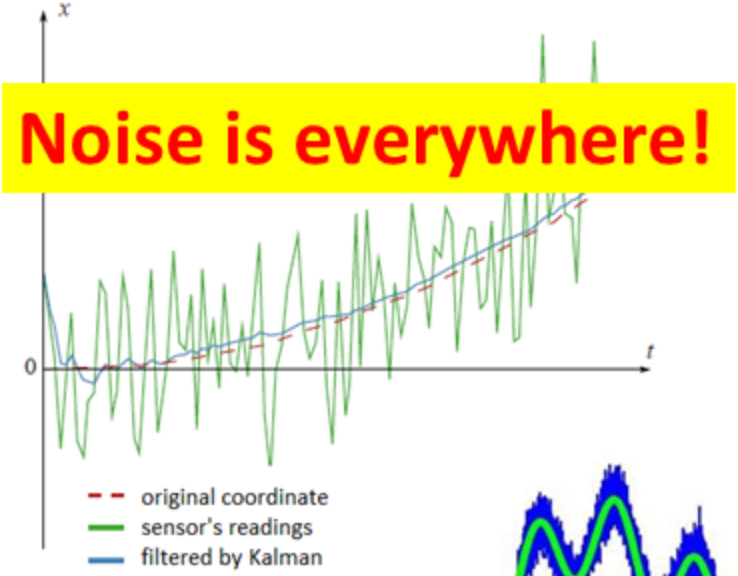
Today’s Agenda

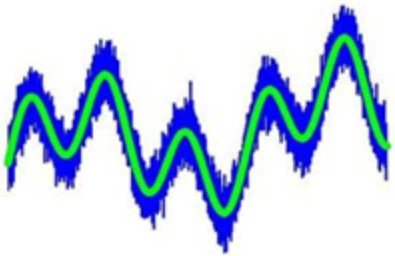
- **What is robot perception? (~12)**
- **Robot vision and computer vision (~5)**
- **Force sensing (~5)**
- **Tactile sensing (~5)**
- **Challenges of robot perception (10)**
- **Algorithms for perception**
  - **State estimation (~5)**
  - **End to end learning (~5)**
  - **Active perception (~5)**
- **Quick Review of Deep Learning (~20)**

Why robot perception is difficult?

1. **Uncertainty**: noise is everywhere!
2. **Modalities**: neural network architectures designed for different sensory modalities
3. **Representations**: representation learning algorithms without strong supervision
4. **Tasks**: state estimation tasks for robot navigation and manipulation
5. **Embodiment**: active perception for embodied visual intelligence

Why do we need a filter?





Engineers use filtering to extract the useful information from noisy signals.

- The main reason to filter a signal is to reduce and smooth out **high-frequency noise** associated with a measurement such as flow, pressure, level or temperature
- When a noisy signal is used in control, filtering is important for effective derivative action and for avoiding excessive movement in the controller output that causes valve wear or disturbs other control loops.
- Ideally, we want to estimate the underlying signal without noise, introducing as little distortion as possible.

https://en.wikipedia.org/wiki/Smoothing

Algorithm	Overview and uses	Pros
Additive smoothing	used to smooth categorical data.	
Butterworth filter	Slower roll-off than a Chebyshev Type I/Type II filter or an elliptic filter	• More linear phase response in the pass-band than Chebyshev Type I/Type II and elliptic filters can achieve. • Designed to have a frequency response as flat as possible in the passband.
Chebyshev filter	Has a steeper roll-off and more passband ripple (type I) or stopband ripple (type II) than Butterworth filters.	• Minimizes the error between the idealized and the actual filter characteristic over the range of the filter
Digital filter	Used on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal	
Elliptic filter		
Exponential smoothing	• Used to reduce irregularities (random fluctuations) in time series data, thus providing a clearer view of the true underlying behaviour of the series. • Also, provides an effective means of predicting future values of the time series (forecasting).[3]	
Kalman filter	• Uses a series of measurements observed over time, containing statistical noise and other inaccuracies by estimating a joint probability distribution over the variables for each timeframe.	Estimates of unknown variables it produces tend to be more accurate than those based on a single measurement alone
Kernel smoother	• used to estimate a real valued function as the weighted average of neighboring observed data. • most appropriate when the dimension of the predictor is low ($p < 3$), for example for data visualization.	The estimated function is smooth, and the level of smoothness is set by a single parameter.

Kalman Filter

motion and sensing discrete-time model for estimation

$$\xi(k) = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \xi(k-1) + \mu$$

$$z(k) = \begin{pmatrix} 1 & 0 \end{pmatrix} \xi(k) + \nu$$

noisy position measure (encoder output)

zero mean Gaussian noises with (co)variances Q (a matrix) and R

T = sampling time

$$\xi(k) = (x(k)\ \dot{x}(k))^T$$

actual state

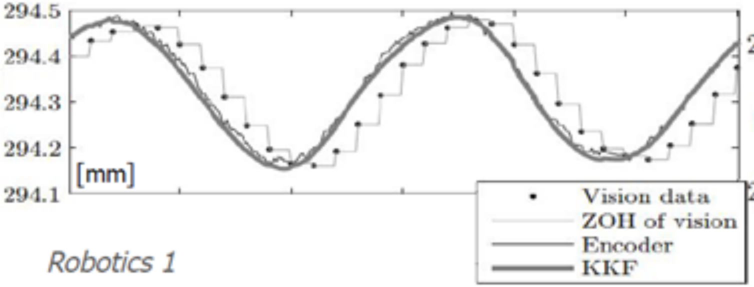
unmeasured velocity

design a (linear) Kalman filter providing an estimate $\hat{\xi}(k)$ of the model state

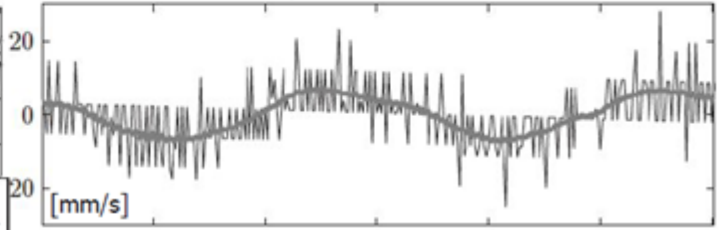
$$\hat{\xi}(k) = \underbrace{\begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \hat{\xi}(k-1)}_{\text{(a priori) prediction}} + K_k \underbrace{\left(z(k) - \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \hat{\xi}(k-1) \right)}_{\text{correction (based on the measured output)}}$$

using the optimal Kalman gain K_k

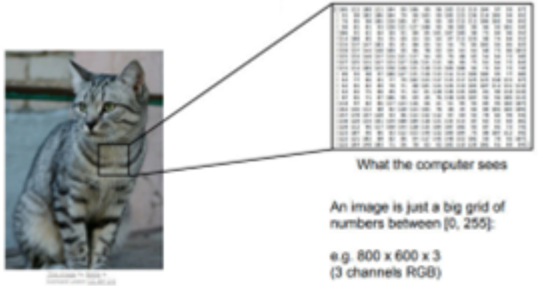
position measure and its filtered version



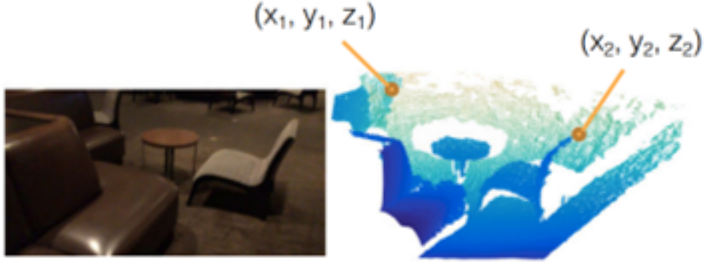
numerical velocity and its filtered estimate



Robot Perception: Modality



Pixels (from RGB cameras)

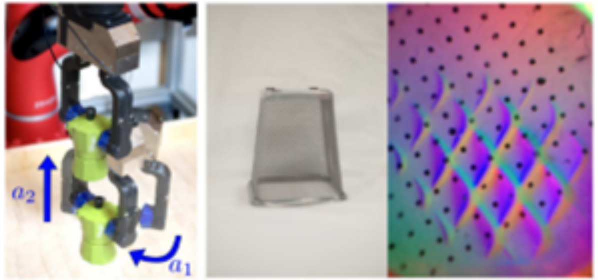


[Source: PointNet++; Qi et al. 2016]

Point cloud (from structure sensors)

[Source: Lee*, Zhu*, et al. 2018]

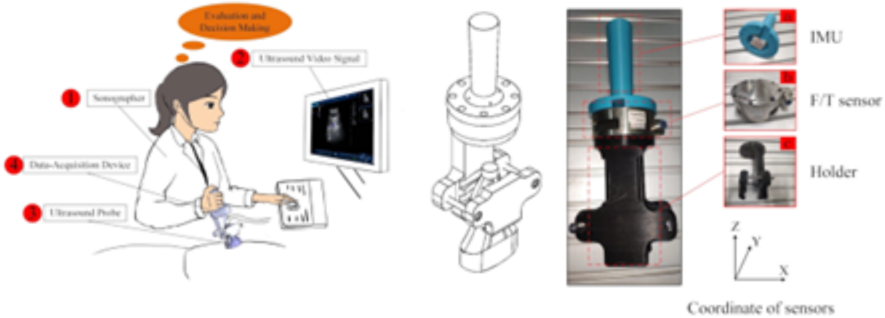
Time series (from F/T sensors)



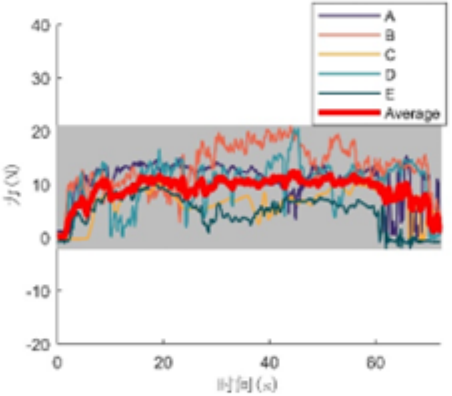
[Source: Calandra et al. 2018]

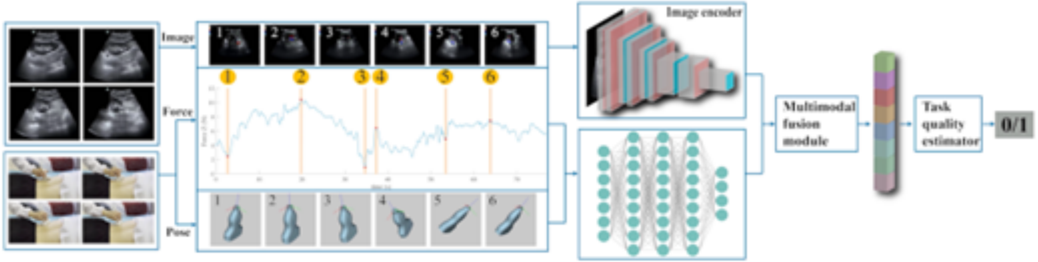
Tactile data (from the GelSights sensors)

Robot Perception: Modality

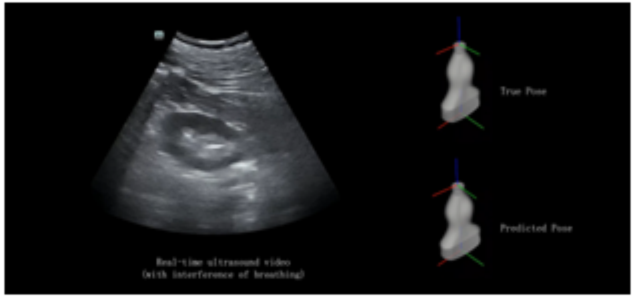


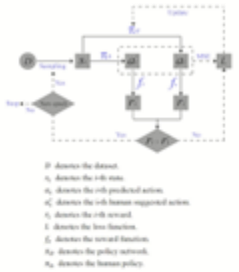
Sonographer ultrasound scanning process CAD model of probe holder Probe holder with sensors

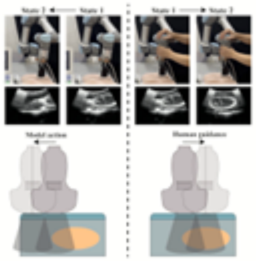




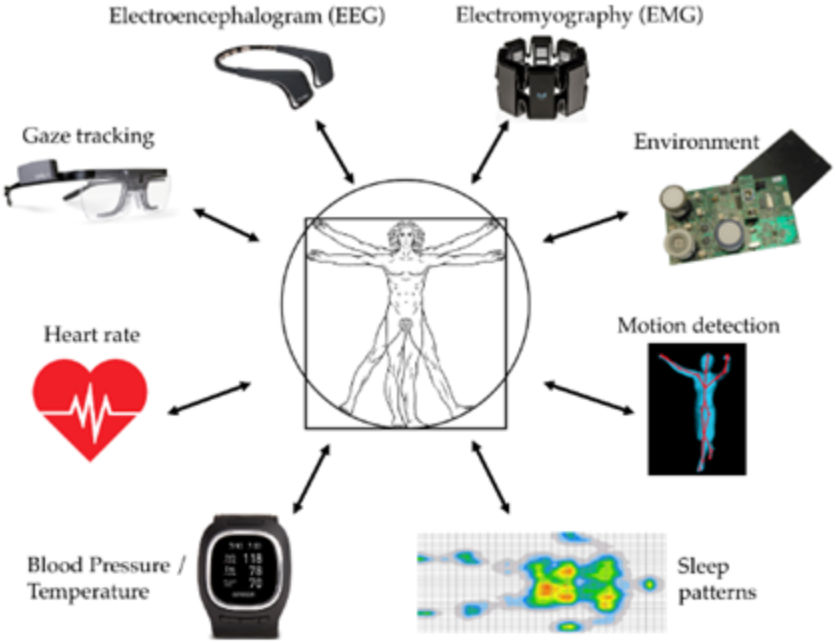
Robot Perception: Modality

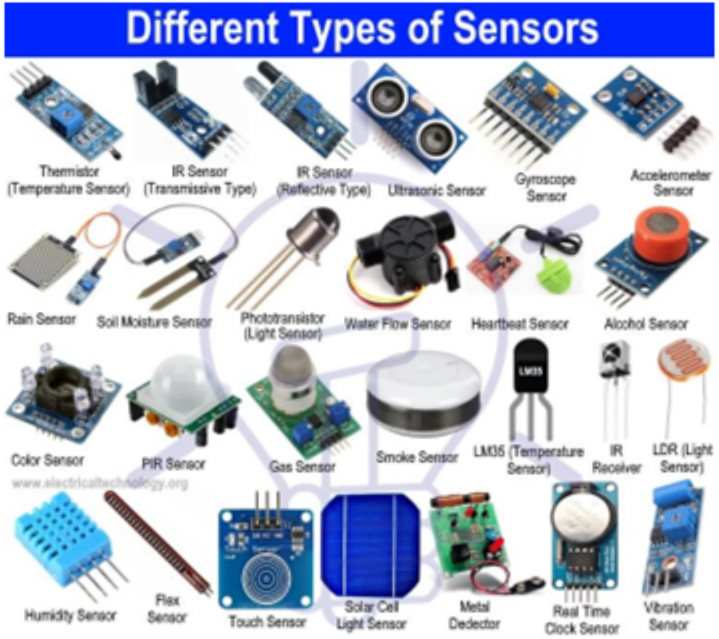




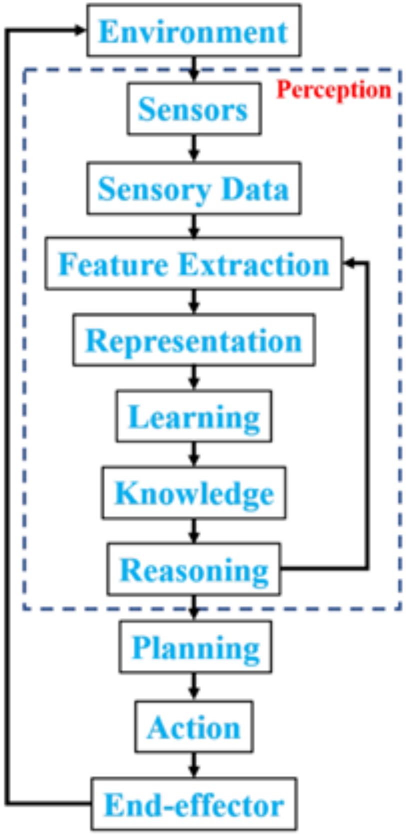


Robot Perception: Modality



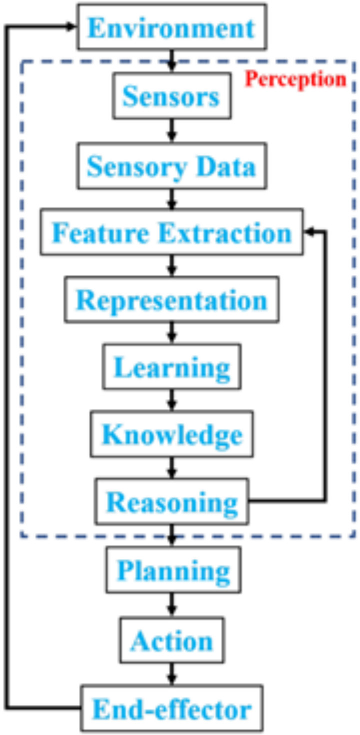


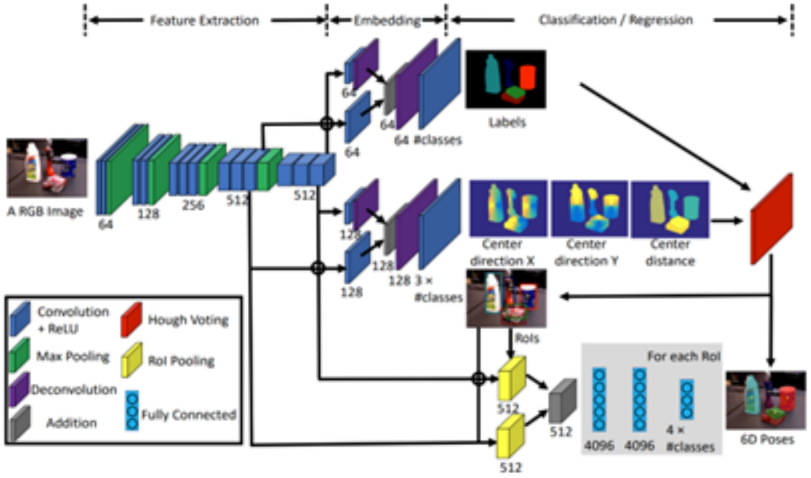
Robot Perception: Modality



- How can we design the algorithms (neural networks) that can effectively process the raw sensory data in different forms?

Robot Perception: Modality

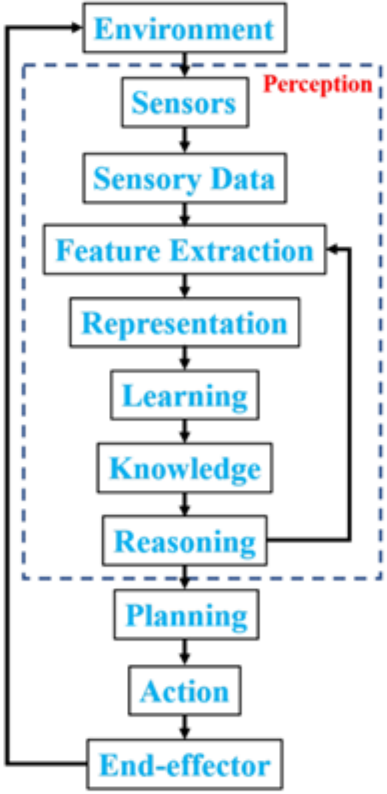


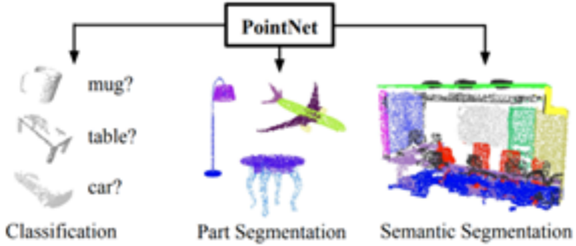


PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes

Yu Xiang[1,2], Tanner Schmidt[2], Venkatraman Narayanan[3] and Dieter Fox[1,2]
[1]NVIDIA Research, [2]University of Washington, [3]Carnegie Mellon University
yux@nvidia.com, tws10@cs.washington.edu, venkatraman@cs.cmu.edu, dieterf@nvidia.com

Robot Perception: Modality





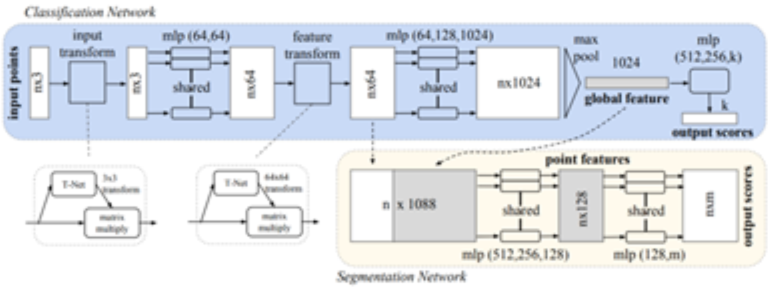


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University

Robot Perception: Modality

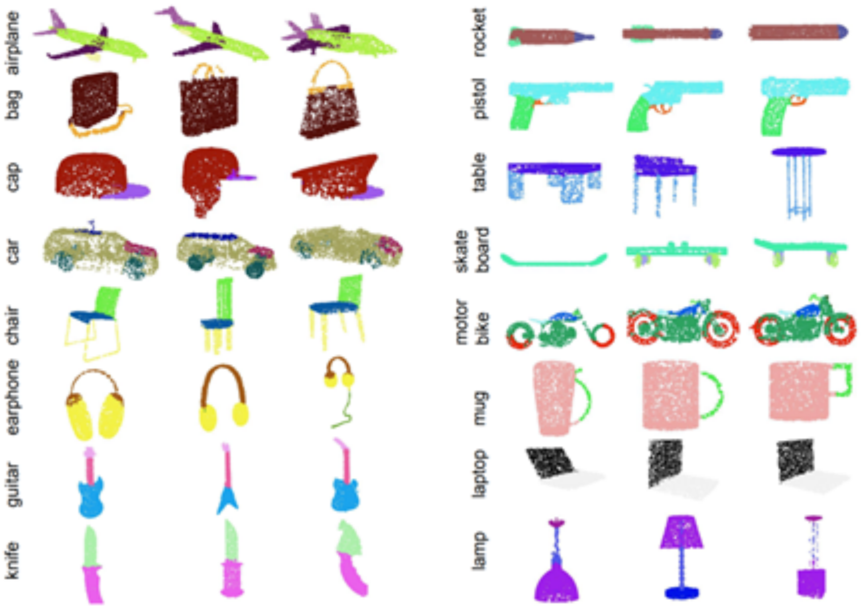
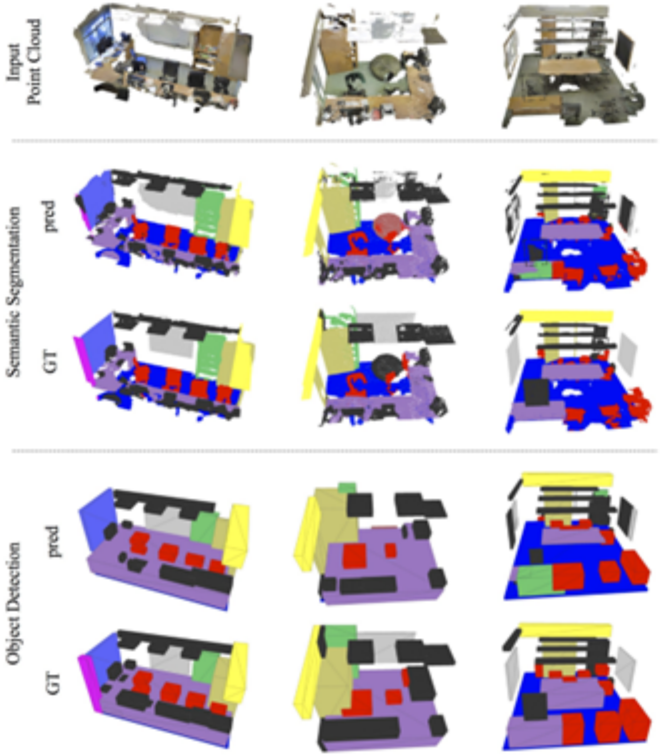
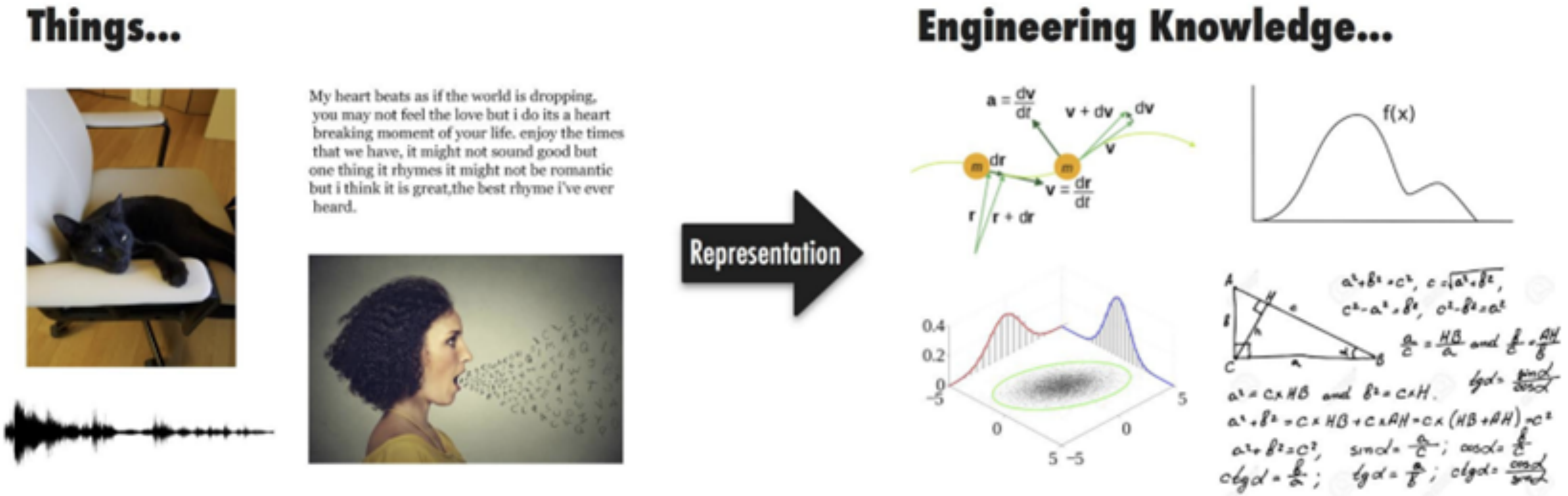


Figure 21. **PointNet segmentation results on complete CAD models.**



Robot Perception: Representation

A fundamental problem in robot perception is to learn the proper **representations** of the unstructured world.

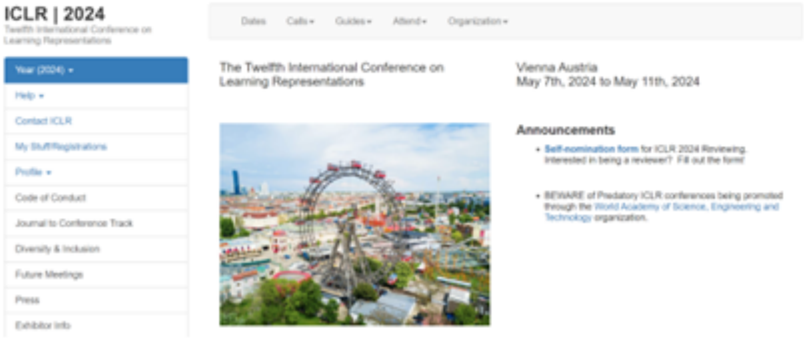


[Source: Stanford CS331b]

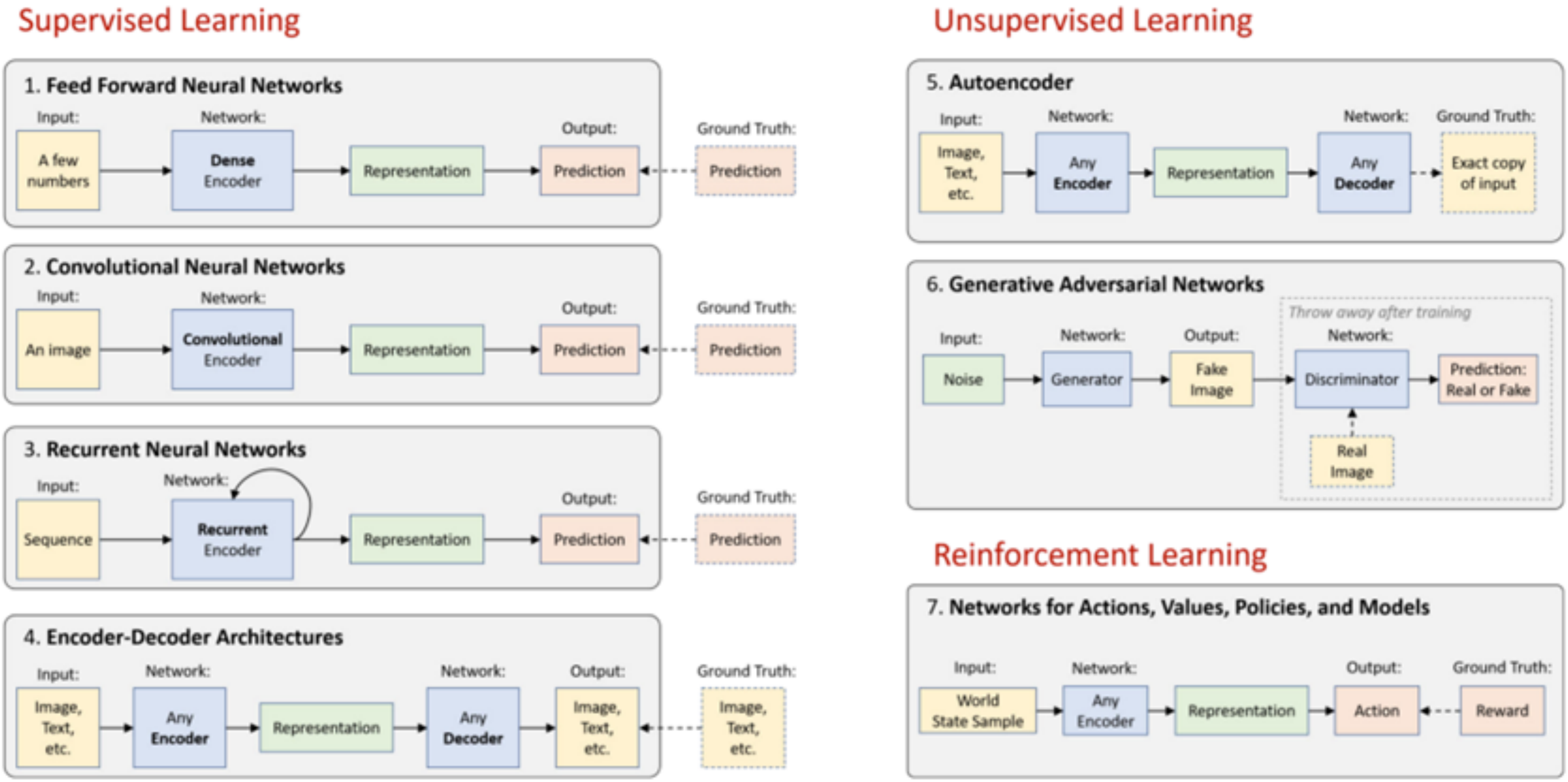
Robot Perception: Representation

"Solving a problem simply means representing it so as to make the solution transparent."

Herbert A. Simon, Sciences of the Artificial



Robot Perception: Representation

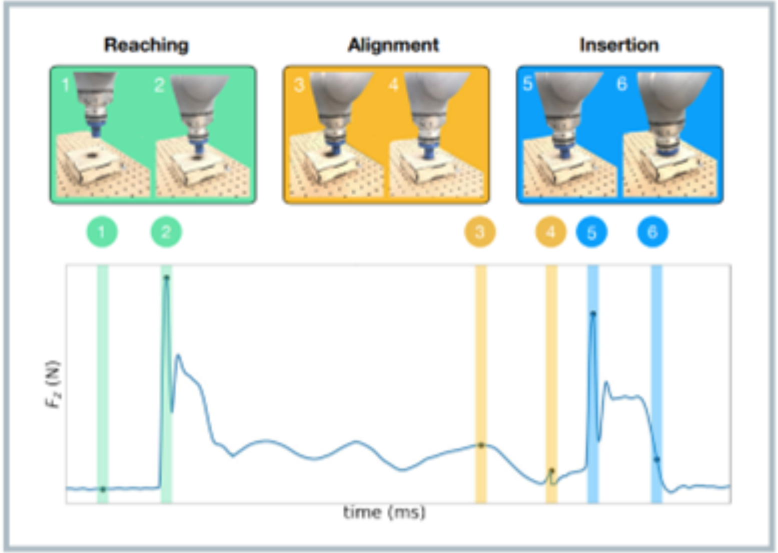


[6.S094, MIT]

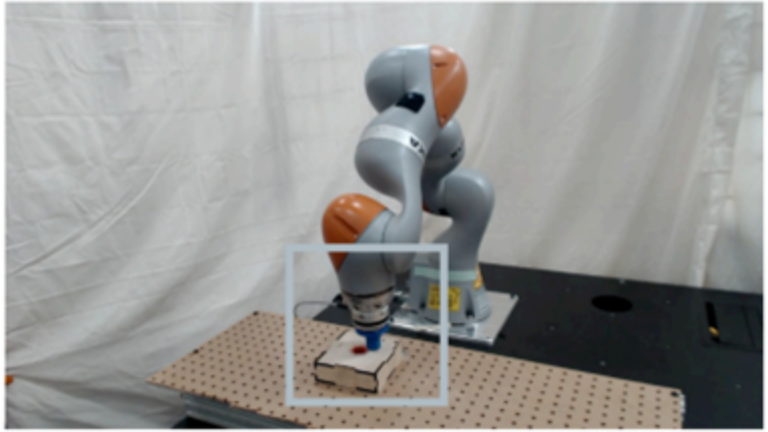
Robot Perception: Representation

How can we learn representations that fuse **multiple sensory modalities** together?

Next course will cover this part

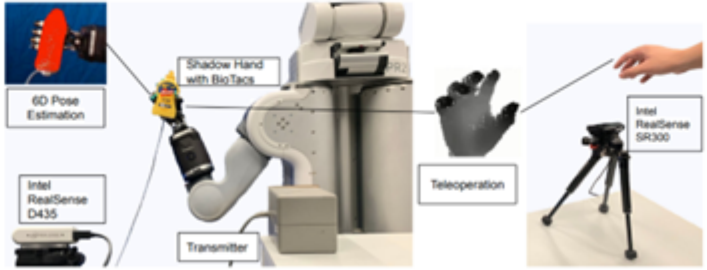


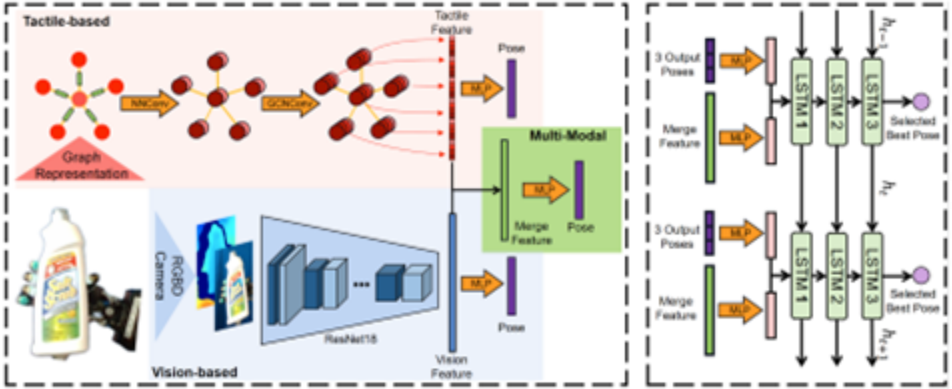
combining vision and force for manipulation



[Lee*, Zhu*, et al. 2018]

Robot Perception: Representation





(a) Three Poses Estimate

(b) SelectLSTM

PoseFusion: Robust Object-in-Hand Pose Estimation with SelectRNN

Yuyang Tu[†1], Junnan Jiang[†2], Shuang Li[1], Norman Hendrich[1], Miao Li[2] and Jianwei Zhang[1]
[1]Universität Hamburg, [2]Wuhan University, †denote equal contribution

PoseFusion: Robust Object-in-Hand Pose Estimation with SelectLSTM

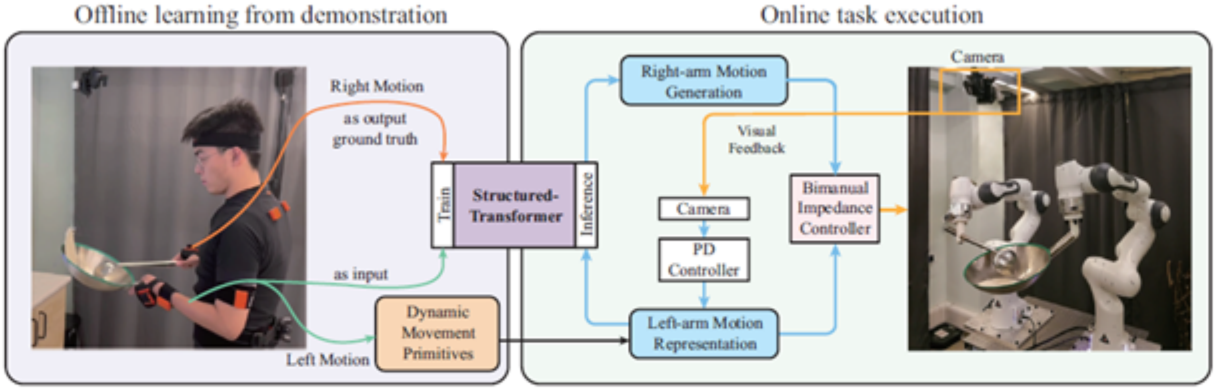
Yuyang Tu[†1], Junnan Jiang[†2], Shuang Li[1], Norman Hendrich[1], Miao Li[*2] and Jianwei Zhang[*1]

与汉堡大学张建伟教授合作（IROS 2023）

Robot Perception: Representation

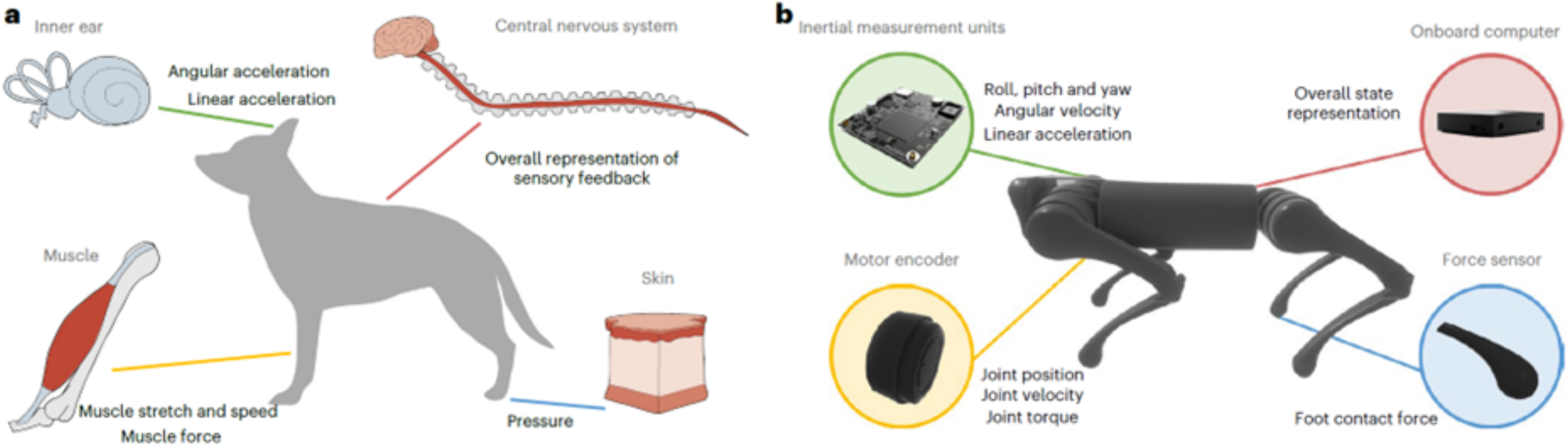
Robot Cooking with Stir-fry: Bimanual Non-prehensile Manipulation of Semi-fluid Objects

Junjia Liu[1], Yiting Chen[1,2], Zhipeng Dong[1], Shixiong Wang[1], Sylvain Calinon[3], Miao Li[†4], and Fei Chen[†1], *Senior Member, IEEE*



与香港中文大学陈翡教授合作（RAL 2022）

Robot Perception: Representation



nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00701-w

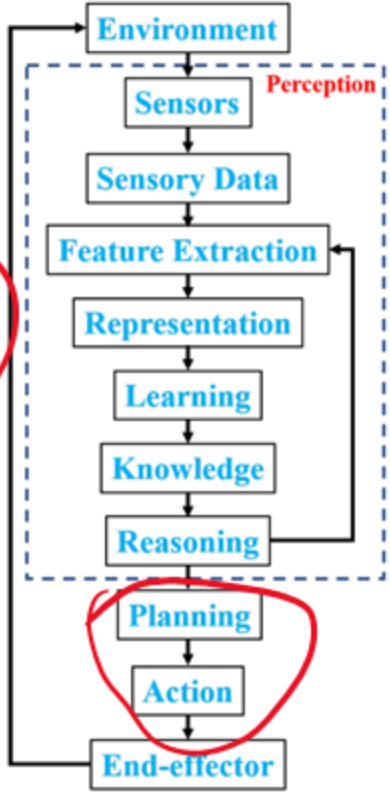
Identifying important sensory feedback for learning locomotion skills

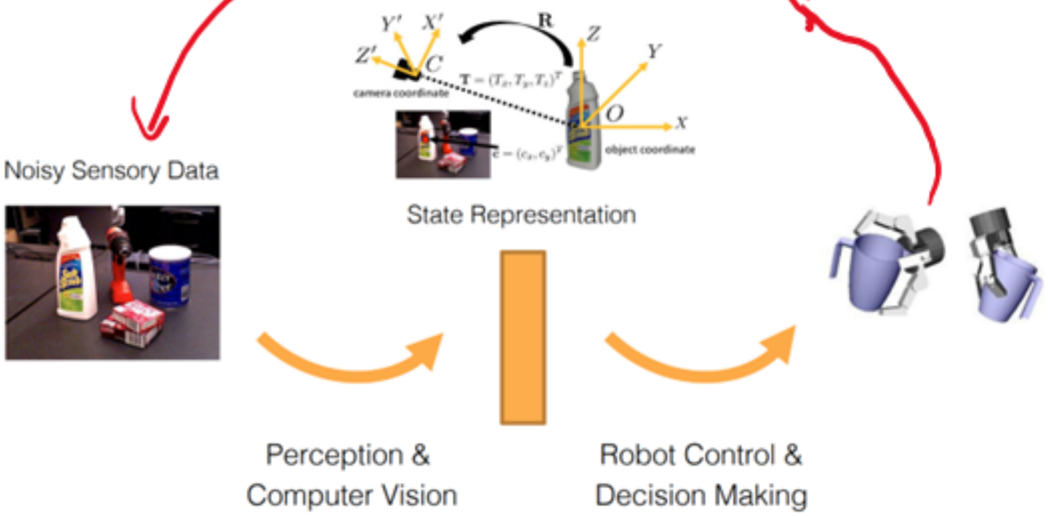
Robot Perception: Task

Task Evaluation.

Task

?





Robotic grasping, manipulation, navigation, exploration, …

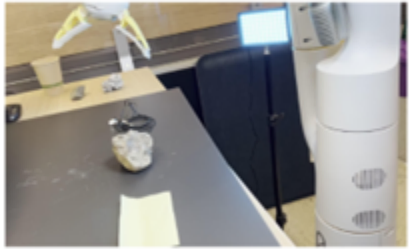
Robot Perception: Task

https://www.deepmind.com/blog/rt-2-new-model-translates-vision-and-language-into-action

Instruction:
I need to hammer a nail, what object from the scene might be useful?

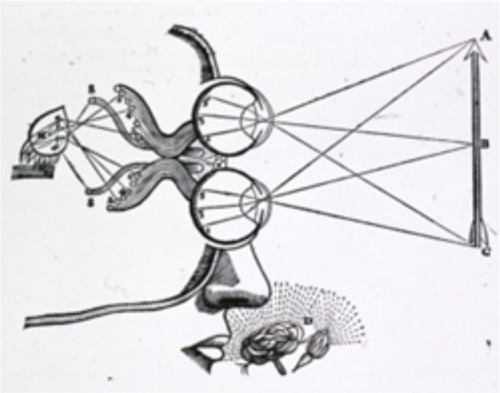
Prediction:
Rocks. Action: 1 129 138 122 132 132 106 127

?



It is in general very difficult to represent a task, human use everyday language to express it.

Robot Perception: Embodiment



Input-Output Picture (Susan Hurley, 1998)

Conventional View of Perception

- Perception is the process of building an internal representation of the environment
- Perception is input from world to mind, and action is output from mind to world, thought is the mediating process.

[Action in Perception, Alva Noë 2004]

"We see in order to move; we move in order to see." – William Gibson

Robot Perception: Embodiment

Pebbles (James J. Gibson 1966)

Embodied View of Perception

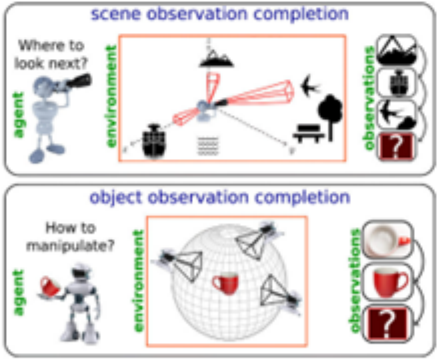
- Subjects asked to find a reference object among a set of irregularly-shaped objects
- Three groups
  a. Passive observers of one static image (49%)
  b. Observers of moving shapes (72%)
  c. Interactive observers (99%)
- The ability to condition input signals with actions is crucial to perception.

Gibson, J. J. (1950). *The Perception of the Visual World*. Oxford England: Houghton Mifflin. ISBN 978-1114828087.

Gibson, J. J. (1966). The senses considered as perceptual systems. Oxford England: Houghton Mifflin. ISBN 978-0313239618.

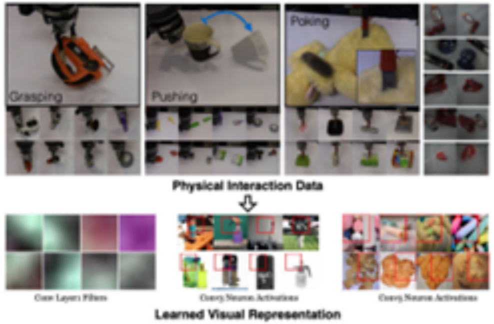
Robot Perception: Embodiment

View Selection



[Ramakrishnan et al. 2019]

Physical Interaction



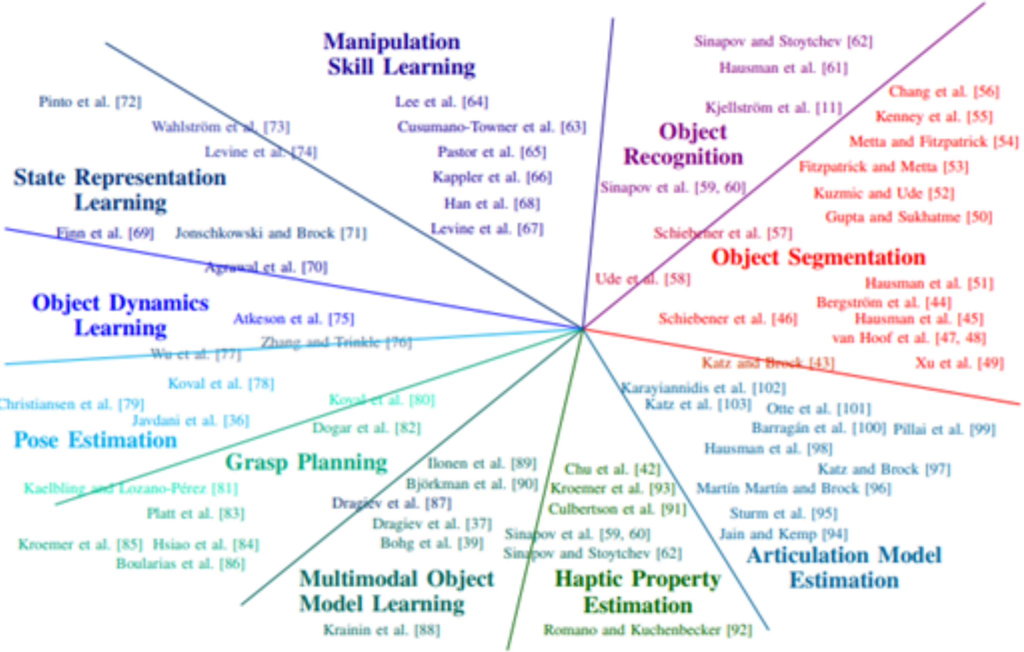
[Pinto et al. 2016]

active perception

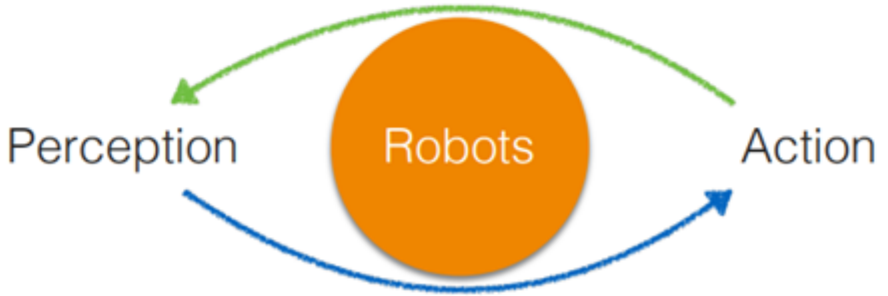
Robot Perception: Embodiment

Interactive Perception: Leveraging Action in Perception and Perception in Action

Jeannette Bohg*, *Member, IEEE,* Karol Hausman*, *Student Member, IEEE,* Bharath Sankaran*, *Student Member, IEEE,* Oliver Brock, *Senior Member, IEEE,* Danica Kragic, *Fellow, IEEE,* Stefan Schaal, *Fellow, IEEE,* and Gaurav Sukhatme, *Fellow, IEEE*



Robot Perception: Embodiment



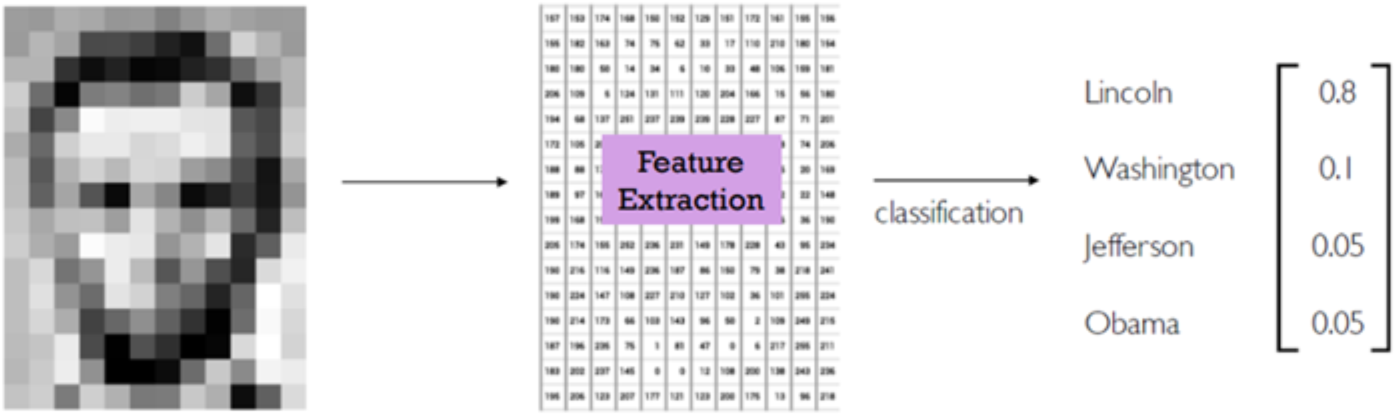
How robots develop better perception from embodied sensorimotor experiences

How robots' intelligent behaviors are guided by their interactive perception

How to close the loop?

Quick review of DL

Tasks in Computer Vision



Input Image

Pixel Representation

- **Regression**: output variable takes continuous value
- **Classification**: output variable takes class label. Can produce probability of belonging to a particular class

Slide Credit: Ava Soleimany, MIT

Quick review of DL

High Level Feature Detection

Let's identify key features in each image category

Nose,
Eyes,
Mouth

Wheels,
License Plate,
Headlights

Door,
Windows,
Steps

Quick review of DL

Manual Feature Extraction

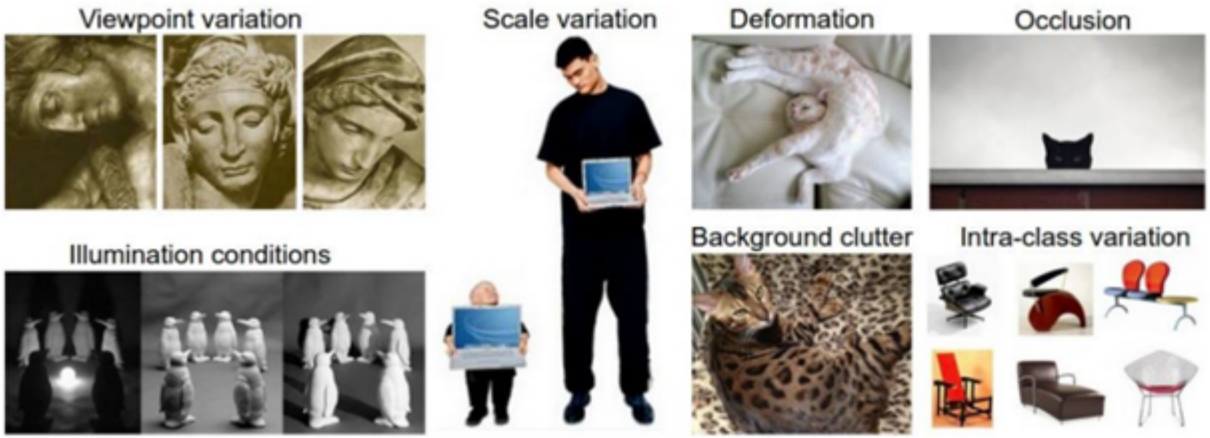
Domain knowledge → Define features → Detect features to classify

Problems?

Quick review of DL

Manual Feature Extraction

Domain knowledge → Define features → Detect features to classify



Quick review of DL

Hand engineered features are time consuming, brittle and not scalable in practice

Question 1 Can we learn the **underlying features** directly from data?

Low Level Features | **Mid Level Features** | **High Level Features**

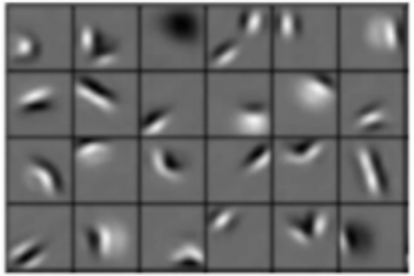
Lines & Edges | Eyes & Nose & Ears | Facial Structure

Quick review of DL

Question 2

Can we **learn hierarchy of features** directly from the data instead of hand engineering?

Low level features



Edges, dark spots

Mid level features

Eyes, ears, nose

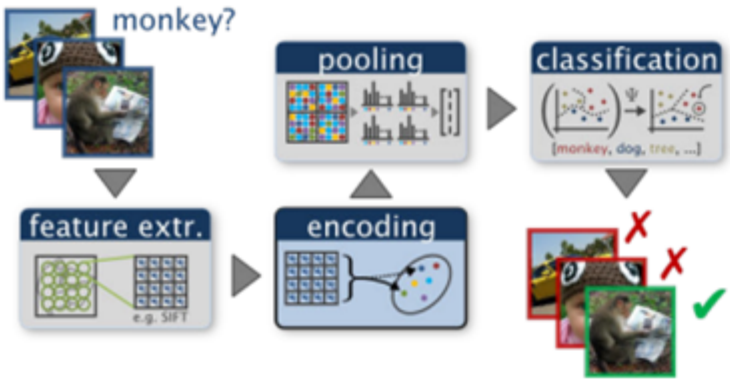
High level features

Facial structure

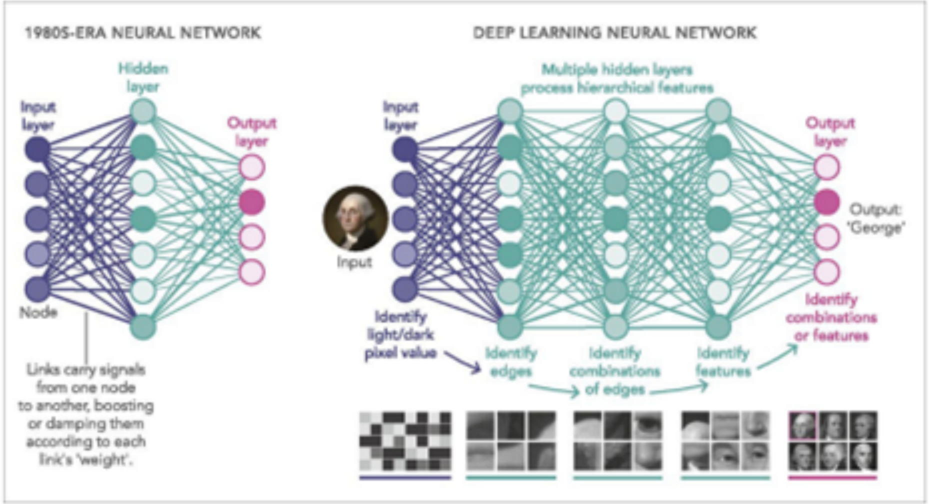
Slide Credit: Ava Soleimany, MIT

Quick review of DL

What is new since 1980s?



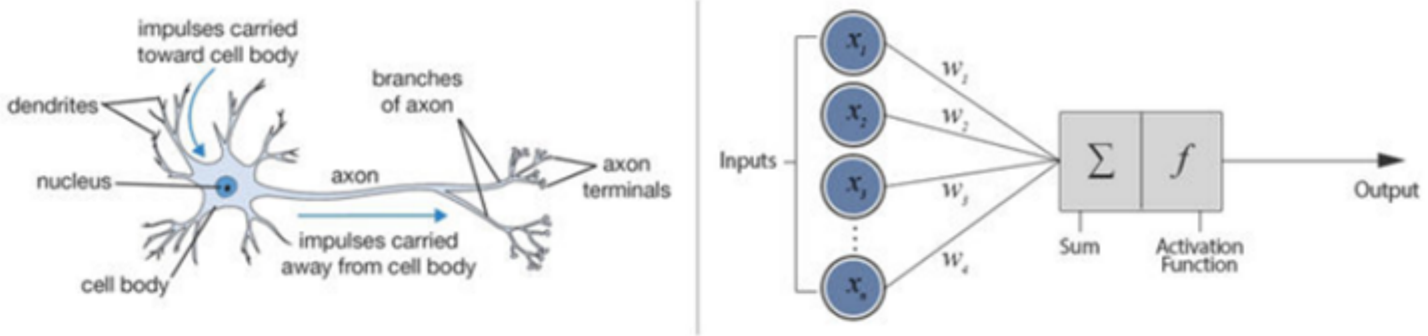
Staged Visual Recognition Pipeline



End-to-end Deep Learning

Quick review of DL

Biological Neuron versus Artificial Neural Network



Biological Neuron

Computational building block for the brain

Artificial Neuron

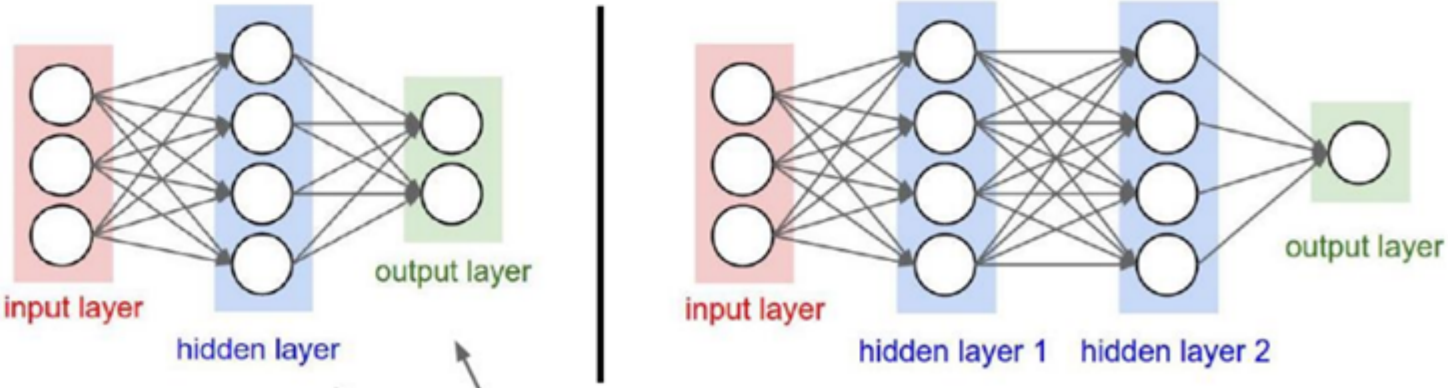
Computational building block for the neural network

Note: Many differences exist – be careful with the brain analogies!

[Dendritic Computation, Michael London and Michael Hausser 2015]

Quick review of DL

Neural Networks: Architectures

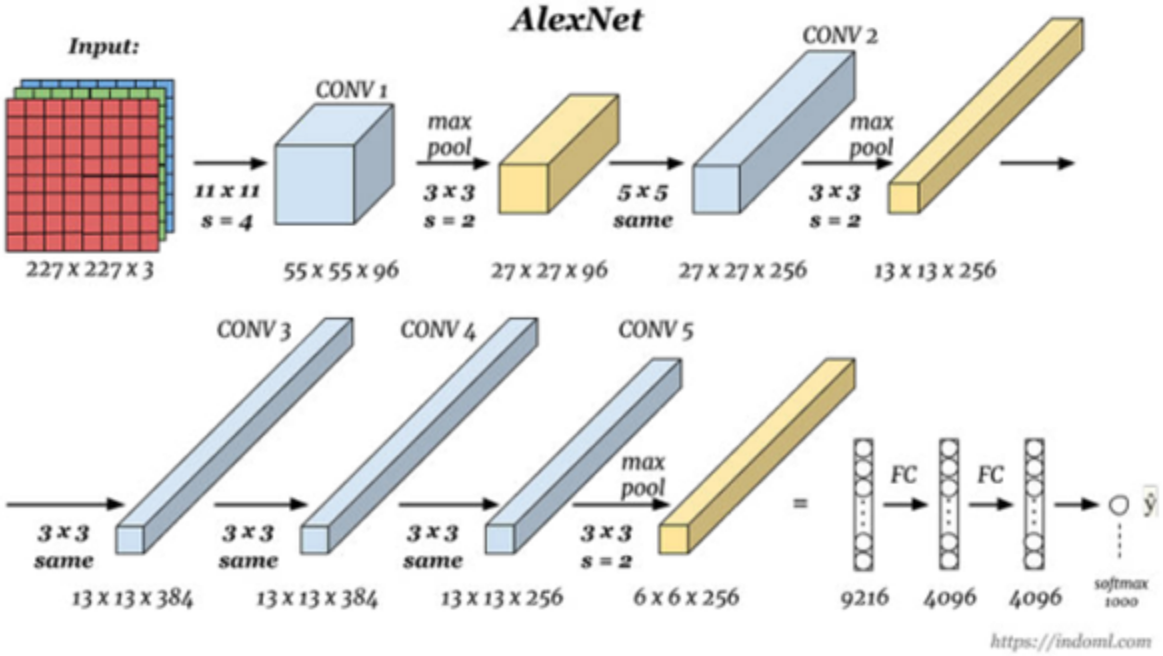


“2-layer Neural Net”, or
“1-hidden-layer Neural Net”

“Fully-connected” layers

“3-layer Neural Net”, or
“2-hidden-layer Neural Net”

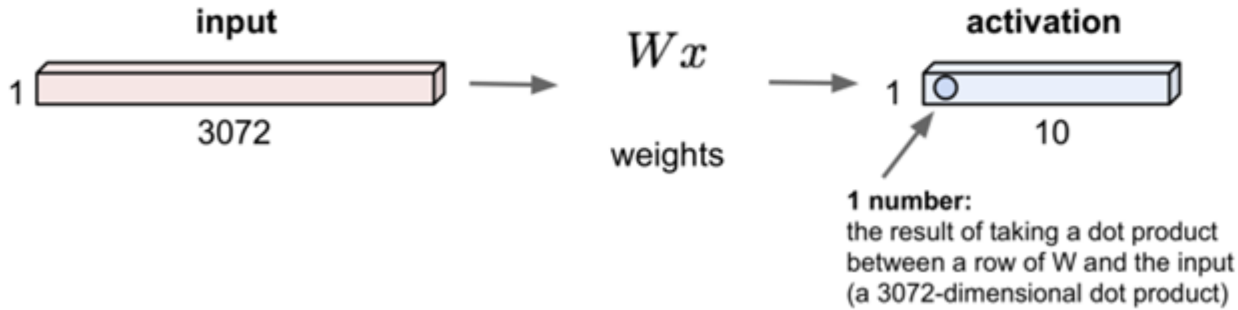
Quick review of DL



Quick review of DL

Fully-Connected Layers

32x32x3 image -> stretch to 3072 x 1

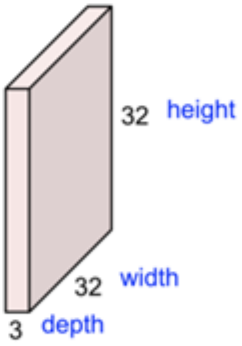


What is the dimension of W?

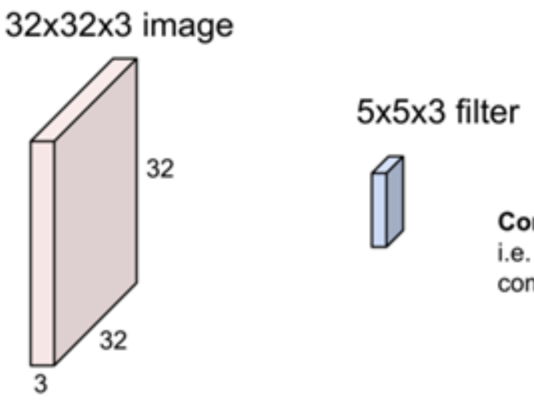
[Source: Stanford CS231N]

Quick review of DL

32x32x3 image -> preserve spatial structure

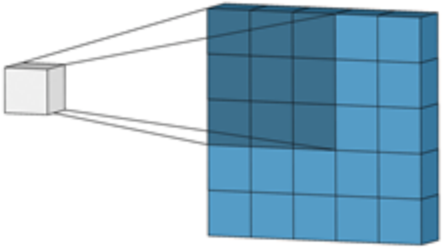


Quick review of DL



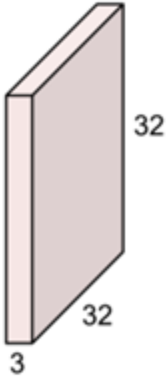
Convolve the filter with the image
i.e. "slide over the image spatially,
computing dot products"

Quick review of DL

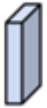


Filters always extend the full depth of the input volume

32x32x3 image

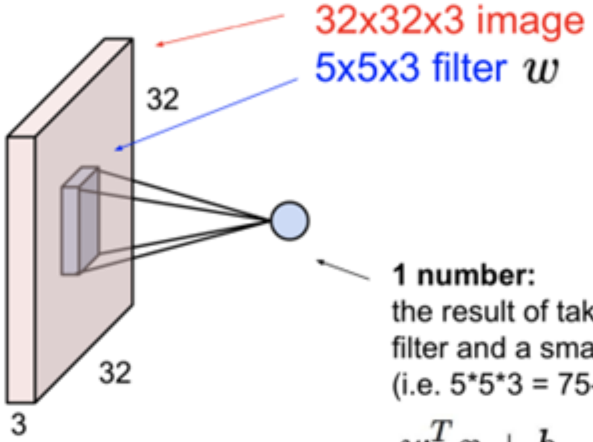


5x5x3 filter



Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

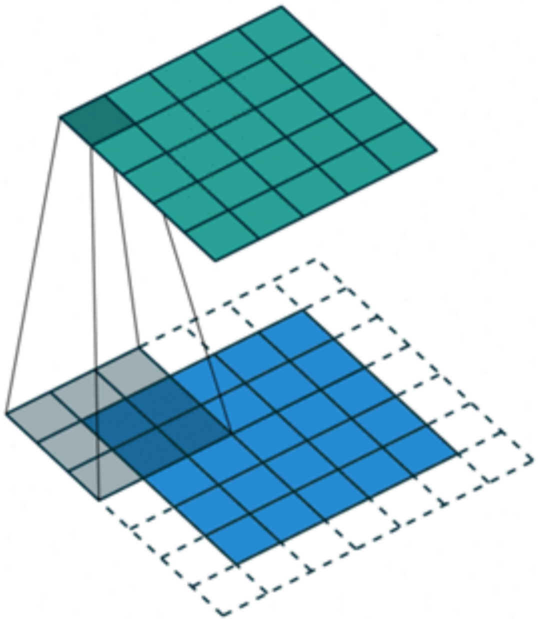
Quick review of DL

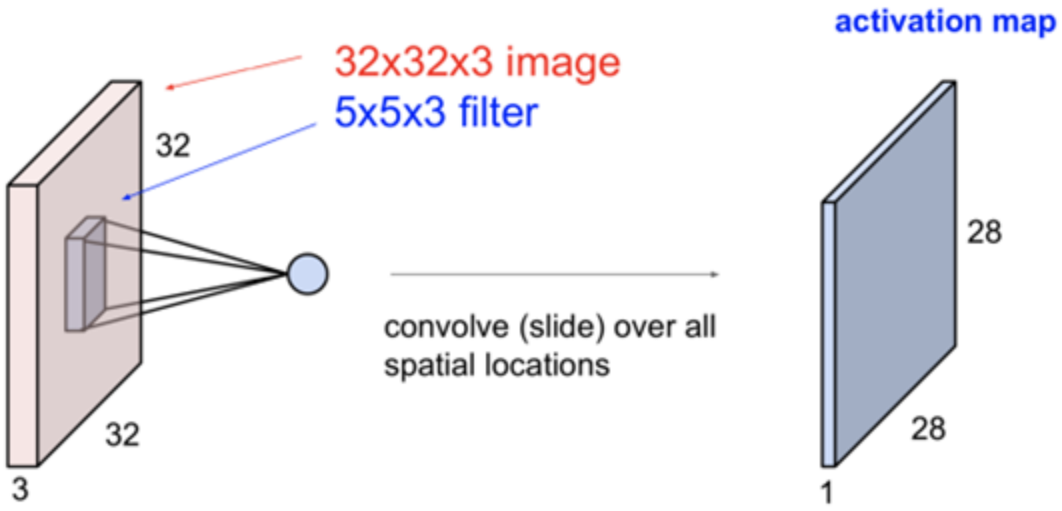


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

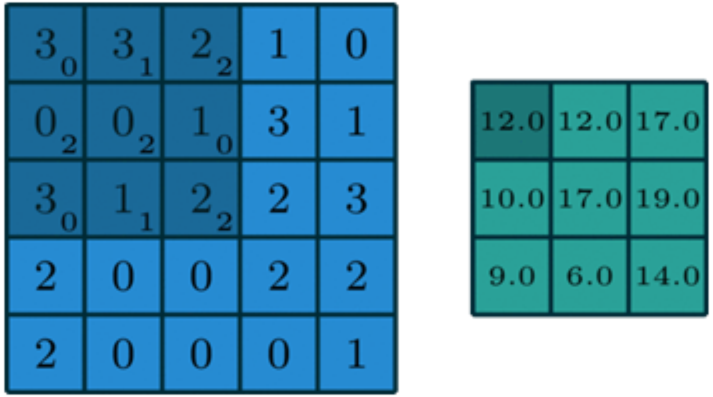
$$w^T x + b$$

Quick review of DL





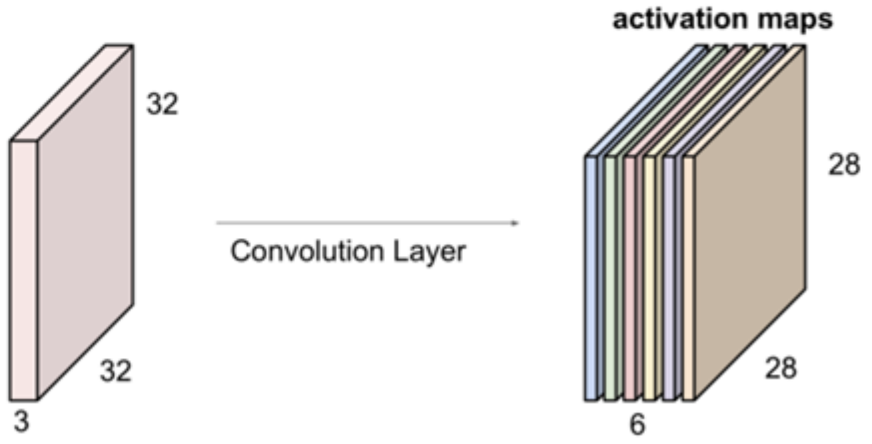
Quick review of DL



consider a second, green filter

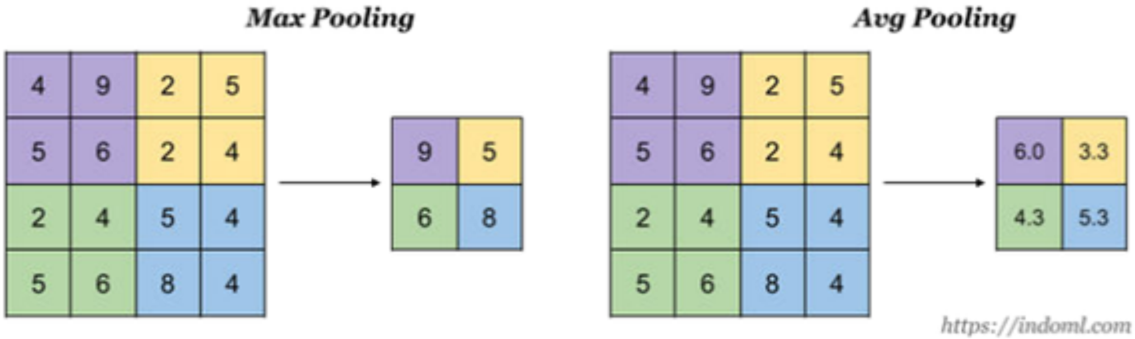
Quick review of DL

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

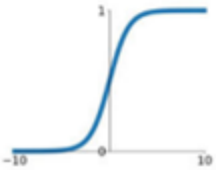
Quick review of DL



Quick review of DL

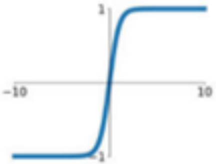
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



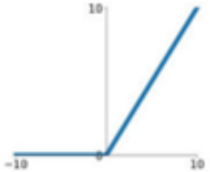
tanh

$\tanh(x)$



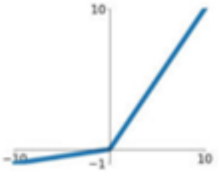
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

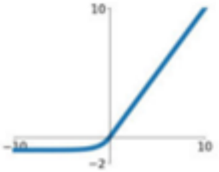


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

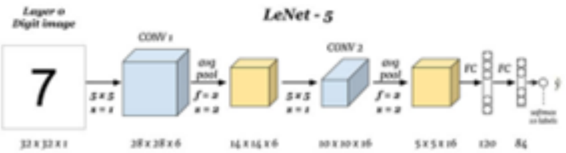
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



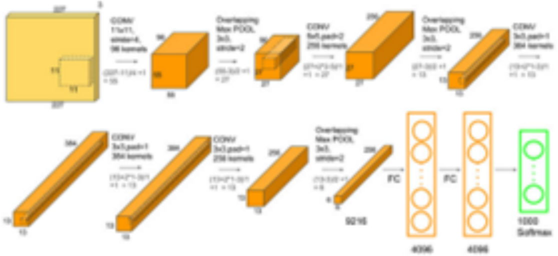
Quick review of DL

LeNet



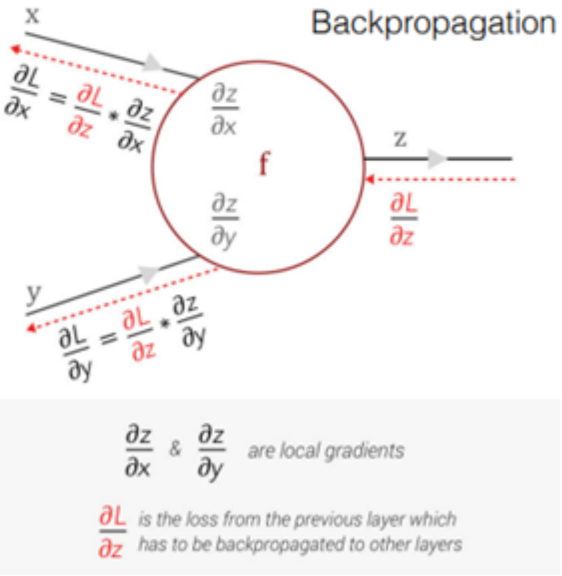
ResNet

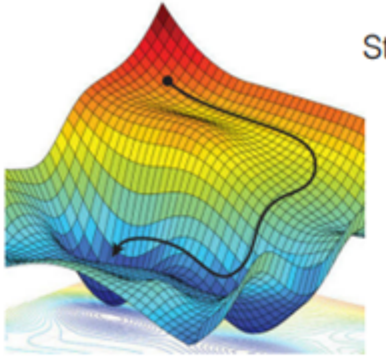
VGG-16



AlexNet

Quick review of DL





Stochastic Gradient Descent (SGD)

learning rate

$$\underbrace{\theta}_{\text{weights}} = \theta - \overbrace{\eta}^{\text{learning rate}} \nabla_\theta J(\theta; \underbrace{x^{(i)}}_{\text{input}}; \underbrace{y^{(i)}}_{\text{label}})$$

weights input label

Quick review of DL

PyTorch

TensorFlow

Tutorial coming in late September / early October

```
[ ] import torch
    from torch import nn

    class MNISTClassifier(nn.Module):

      def __init__(self):
        super(MNISTClassifier, self).__init__()

        # mnist images are (1, 28, 28) (channels, width, heig
        self.layer_1 = torch.nn.Linear(28 * 28, 128)
        self.layer_2 = torch.nn.Linear(128, 256)
        self.layer_3 = torch.nn.Linear(256, 10)

      def forward(self, x):
        batch_size, channels, width, height = x.size()

        # (b, 1, 28, 28) -> (b, 1*28*28)
        x = x.view(batch_size, -1)

        # layer 1
        x = self.layer_1(x)
        x = torch.relu(x)

        # layer 2
        x = self.layer_2(x)
        x = torch.relu(x)

        # layer 3
        x = self.layer_3(x)

        # probability distribution over labels
        x = torch.log_softmax(x, dim=1)

        return x
```

Quick review of DL

Online Courses

- CS231N: Convolutional Neural Networks for Visual Recognition
  http://cs231n.stanford.edu/
- MIT 6.S191: Introduction to Deep Learning
  http://introtodeeplearning.com/

Textbooks:

- Deep Learning. Ian Goodfellow, Yoshua Bengio, Aaron Courville
  http://www.deeplearningbook.org/