Introduction to Computer Vision
VIC-Introduction to Visual Computing

Course built by Maria Vakalopoulou
Teached by Céline hudelot
Mathematics and Informatics (MICS)
CentraleSupelec, University Paris-Saclay

About Maria

- Undergrad at National Technical University of Athens, Greece
- Ph.D in CS and CV at National Technical University of Athens, Greece
- PostDoc researcher at CentraleSupelec
- Assistant Professor at CentraleSupelec (since January 2019)

Research interests: Computer Vision, Machine Learning, Deep Learning, Medical Imaging and Remote Sensing

About Me

- Ph.D in CS and CV at INRIA Sophia-Antipolis
- PostDoc researcher at Telecom Paris-Tech
- Assistant Professor at Centrale and Professor at CentraleSupélec
- Head of the MICS Laboratory
- **Research interests:** AI for non-structured data interpretation, machine learning, knowledge representation and reasoning.

Prerequisites

- Linear Algebra ~70%
- Numerical mathematics - optimization ~60%
- Statistics and probability ~50%

- Hands on:
  - Python programming
  - OpenCV, Scikit-learn, ….

Outline of the Course

- 1h30 Lecture and 1h30 Practical Session
- Interactions via edunao!
  - Material, Assignments
  - Announcements

- Content: ***[NOT DEEP LEARNING – Some connections explained]***
  - Low vision (feature extraction, edge detection)
  - Stereo Vision and Optical Flow
  - Segmentation and Grouping
  - Object detection and Tracking

Outline of the Course

- The evaluation of the course will be based on: **[55%]**
  - Two assignments: the assignments will include theoretical questions as well hands-on practical questions.

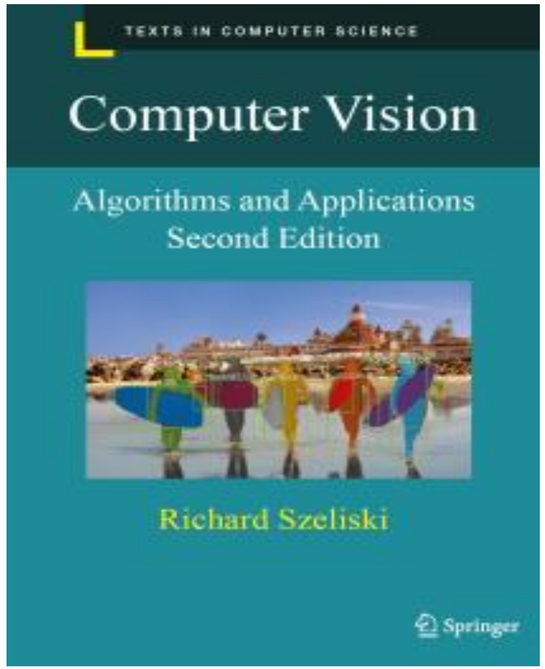
- Project: you are expected to form groups of 2-3 people, propose a topic for your project, and submit a final project report. **[45%]**
  - Open project. ***[NOT DEEP LEARNING]***

Outline of the Course

- Bibliography:
  - Antonio Torralba, Phillip Isola and William T. Freeman – Foundation of Computer Vision
    - https://mitpress.mit.edu/9780262048972/foundations-of-computer-vision/
  - Rick Szeliski, Computer Vision: Algorithms and Applications
  - David Forsyth and Jean Ponce, Computer Vision: A modern Approach
    - See : http://luthuli.cs.uiuc.edu/~daf/book/bookpages/pdf/front.pdf
  - Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing
    - Foundations of image processing

Outline of the Course

- Bibliography:
  - Rick Szeliski, Computer Vision: Algorithms and Applications
  - http://szeliski.org/Book/



Recent work, with a strong focus on applications.

Outline of the Course

- Bibliography:
  - **Antonio Torralba, Phillip Isola and William T. Freeman – Foundation of Computer Vision**
  - **Rick Szeliski, Computer Vision: Algorithms and Applications**
  - David Forsyth and Jean Ponce, Computer Vision: A modern Approach
  - Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing
- A lot of scientific papers
  - CVPR, ICCV, ECCV, ACCV, BMVC, ICPR, NeurIPS, MICCAI, ISBI, …
  - PAMI, IJCV, CVIU, MEDIA, Pattern Recognition, ….
- For the deep learning part :
  - Dive into Deep Learning, Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola (https://d2l.ai/)

Roadmap

- Motivation and Applications of Vision
- A brief history of Vision
- Main approaches in Vision
- Linear Algebra Prerequisites

What is computer vision?

Image from the CamVid Dataset

What is computer vision?

Image from the CamVid Dataset

Where are the cars?

What is computer vision?

Image from the CamVid Dataset

Where are the cars?

What is computer vision?

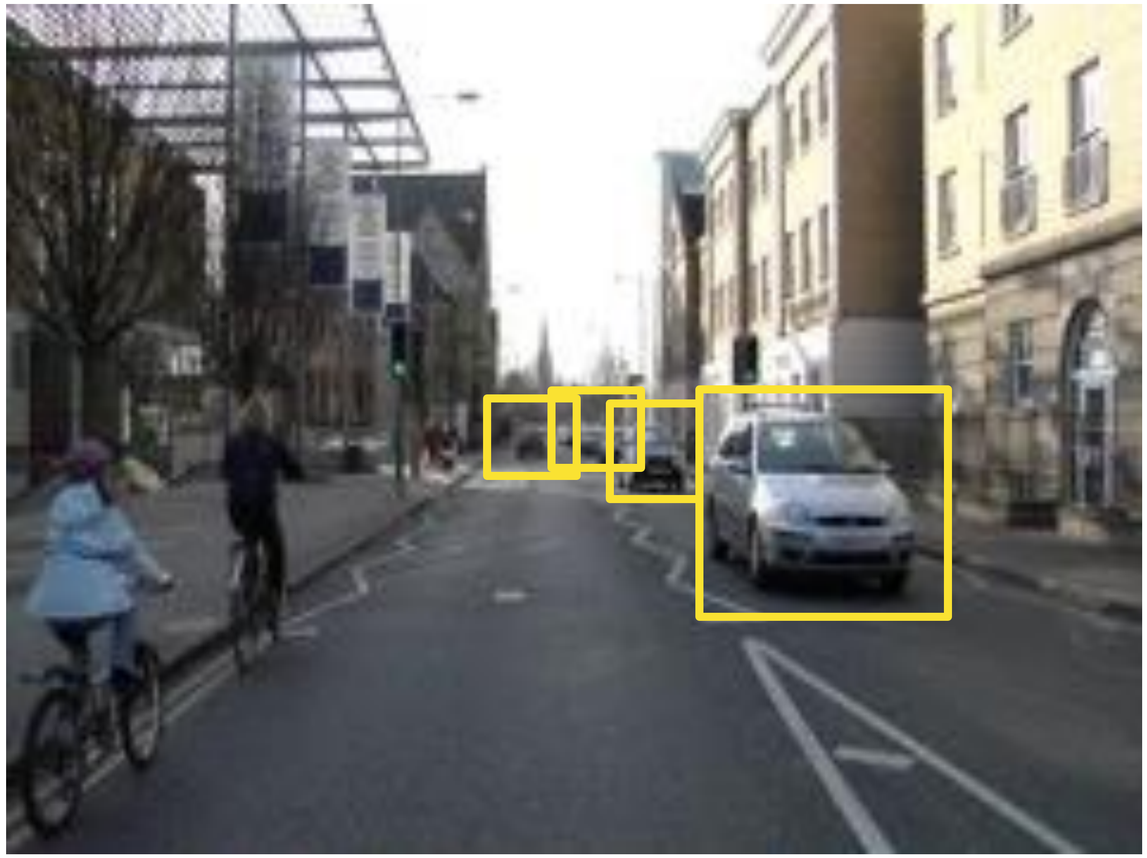
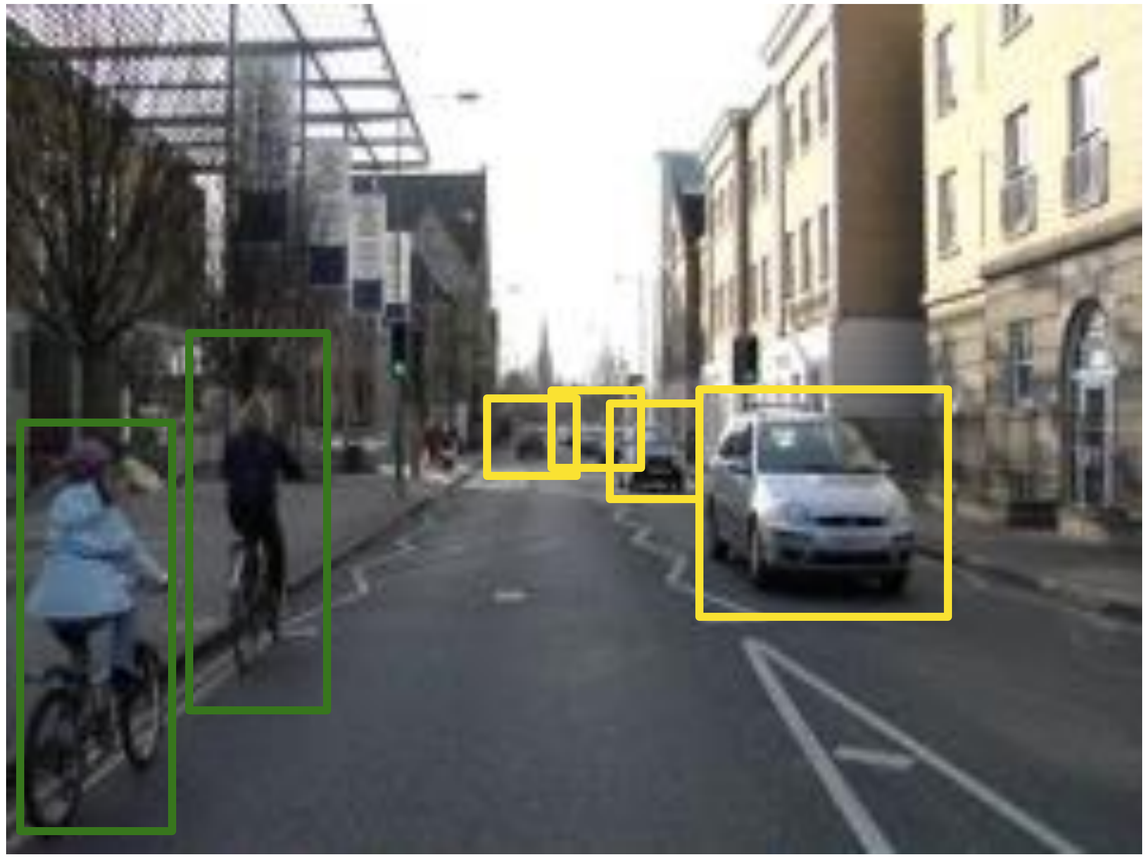


Image from the CamVid Dataset

Where are the cars?

What is computer vision?



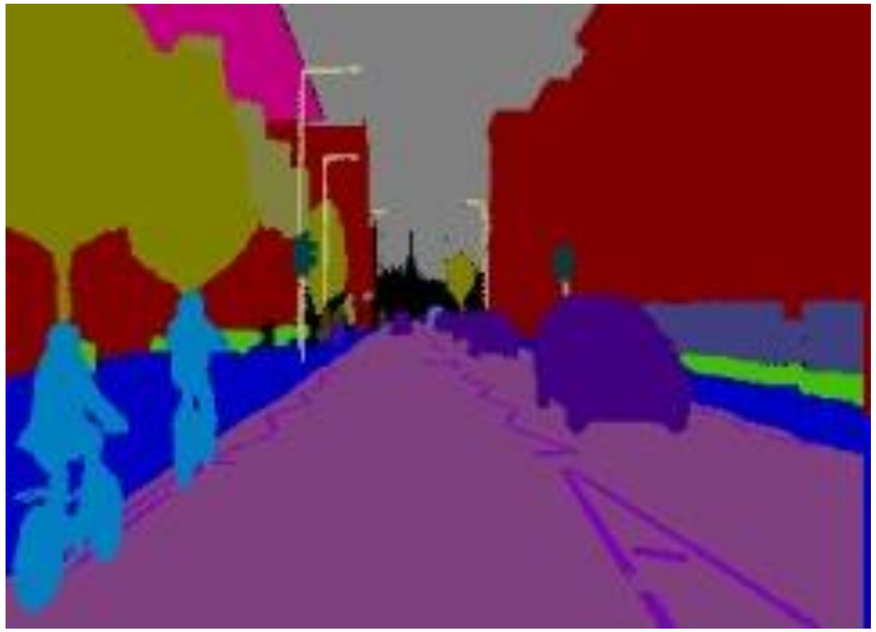
Where are the cars?
Where are the people?

Image from the CamVid Dataset

What is computer vision?

Computer Vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos.

Image from the CamVid Dataset

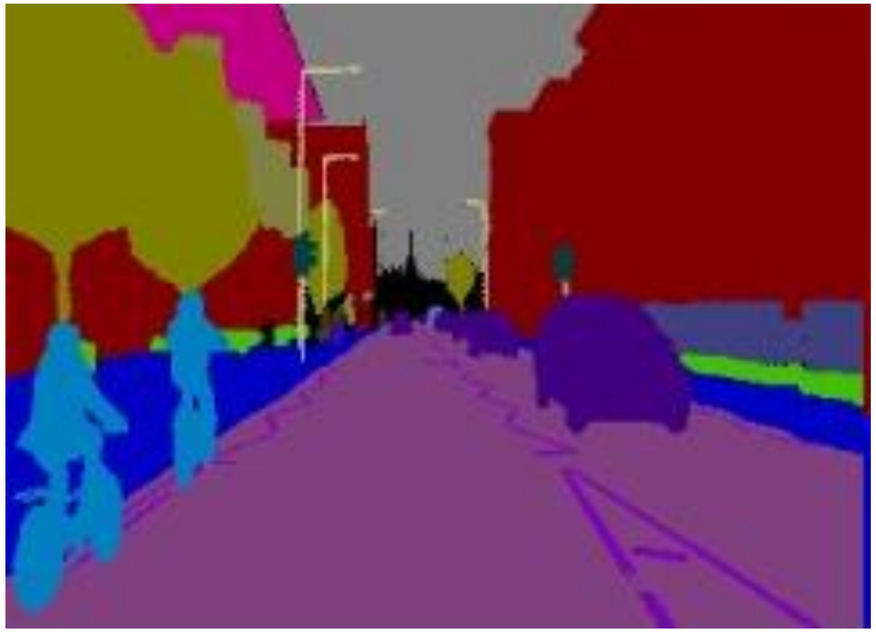


Corresponding ground truth image

What is computer vision?

- **Computer Vision** is difficult. It is an inverse problem: recover some unknowns given insufficient information to fully specify the solution.

Image from the CamVid Dataset



Corresponding ground truth image

What is computer vision?

- **Computer Vision** is difficult. It is an inverse problem: recover some unknowns given insufficient information to fully specify the solution.

What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

Reduce the gap between the pixels and their meanings.

FIGURE – Source : Fei-Fei Li, Computer Vision Course, Stanford

What is computer vision ?

- Trucco & Verri : *computing properties of the 3D world from one or more digital images.*
- Stockman & Shapiro : *to make useful decisions about physical objects and scenes based on sensed images.*
- Ballard & Brown : *the construction of explicit, meaningful descriptions of physical objects from images.*
- Forsyth & Ponce : *extracting descriptions of the world from pictures or sequences of pictures.*
- Szeliski : *write computer programs that can interpret images.*

What is computer vision ?

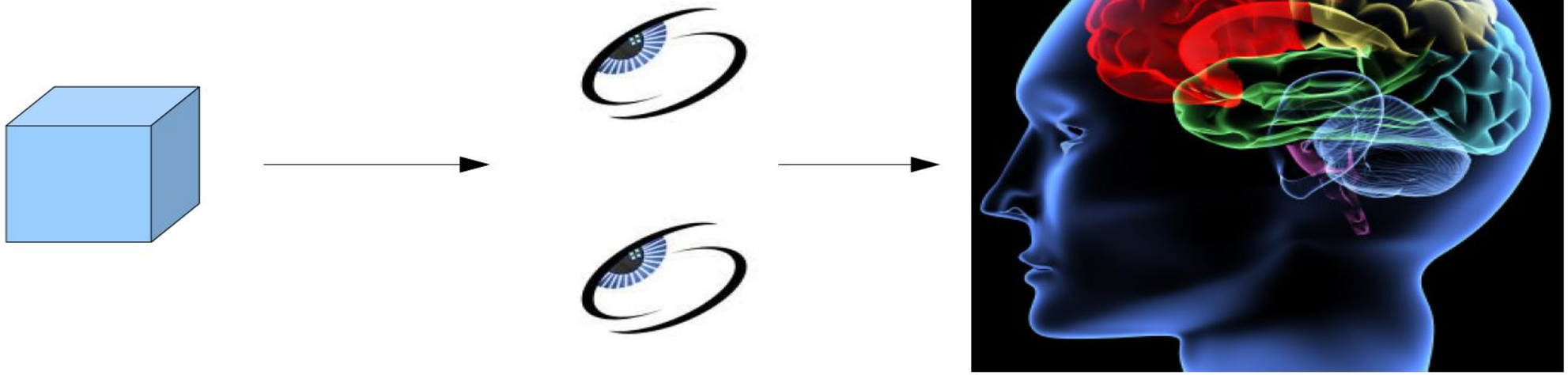
- Computer Vision (CV) is a field of Artificial Intelligence (AI) that deals with computational methods to help computers understand and interpret the content of digital images and videos.
- Computer vision aims to make computers see and understand visual data input from cameras or sensors.

Human Vision

3D scene

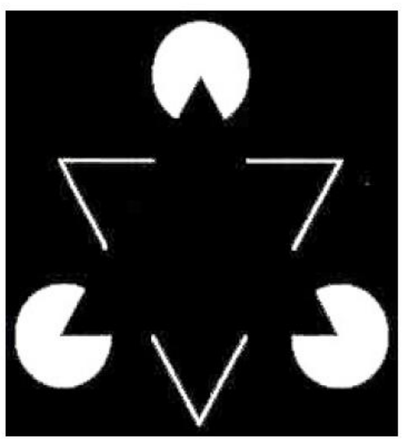
Sensor

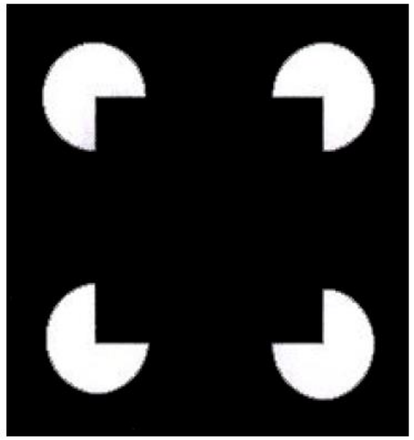
Interpretation



Human Vision: imperfections

The human brain prefers a structured information

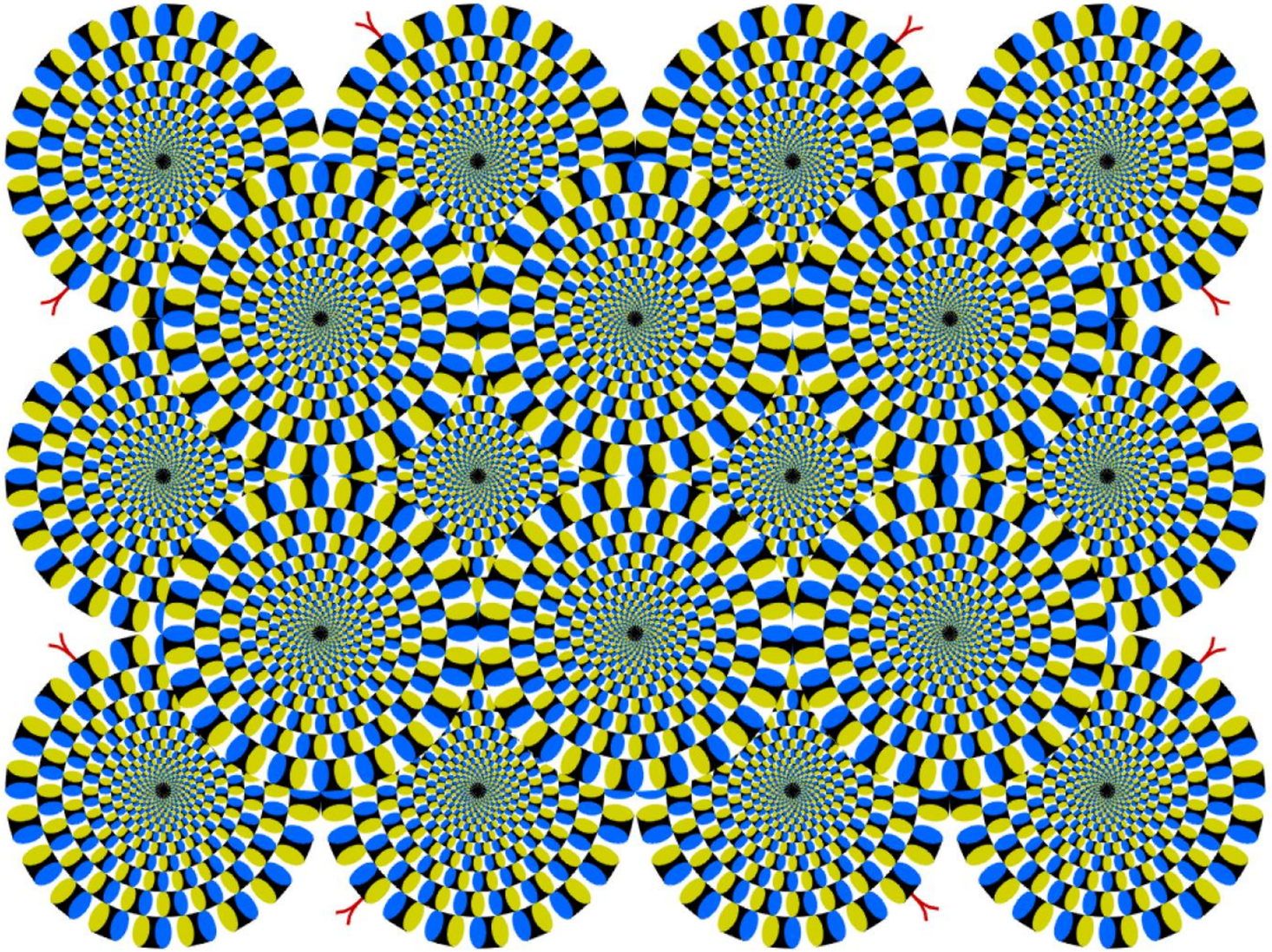




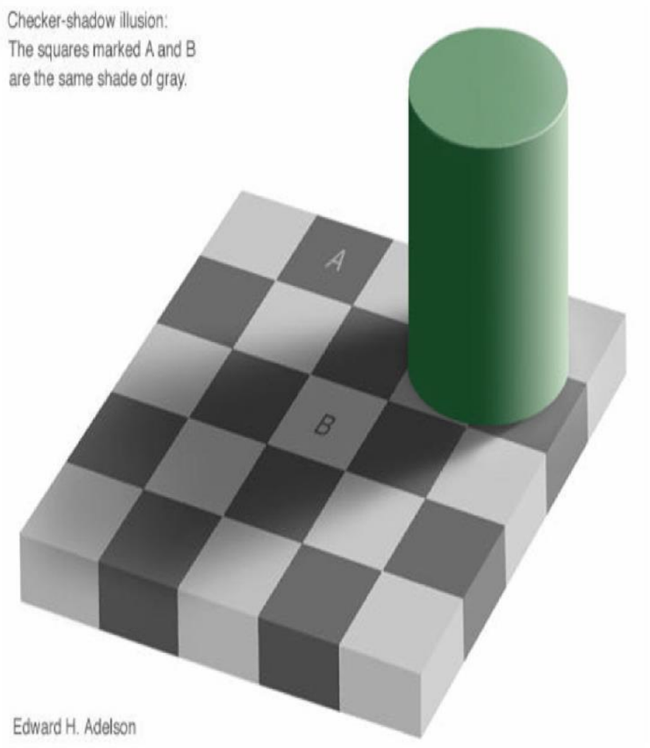
Non existing edges

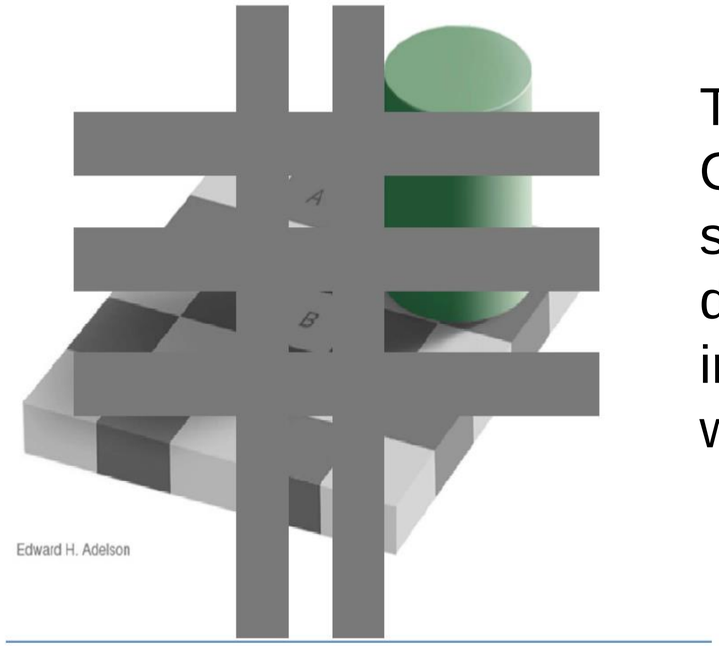
https://michaelbach.de/ot/index.html
https://www.illusionsindex.org/i
https://www.ritsumei.ac.jp/%7Eakitaoka/index-e.html

Human Vision: imperfections



Human Vision: imperfections





The squares have the same color. Our brains force us to imagine the squares as they should be: one dark and the other light. The importance of perception in the way we interpret the world.

Human Vision: imperfections

Change Blindness

https://sites.socsci.uci.edu/~ddhoff/cb.html

http://nivea.psycho.univ-paris5.fr/RensinkOReganClarkVisCog/RensORClark.pdf

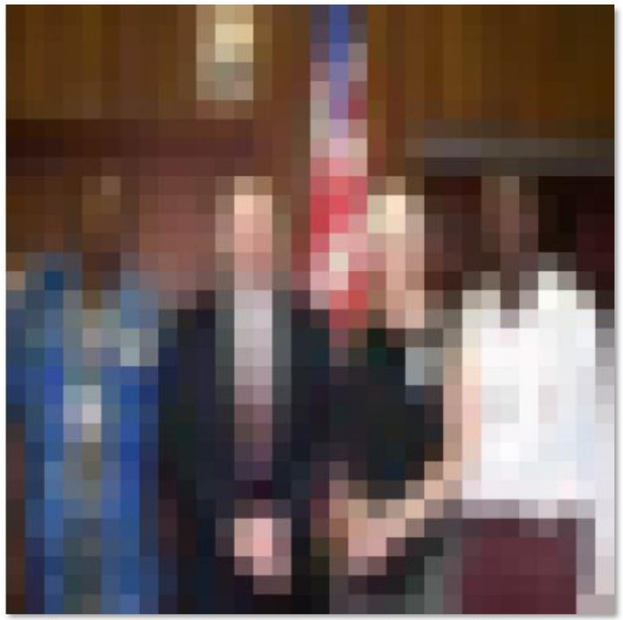
Human Vision : high performing in recognition

Human vision is imperfect but human vision system is very good in recognition : Image interpretation: very easy and very fast for a human [Thorpeet al., Nature 1996] 3 : reco animal < 150 ms

https://pubmed.ncbi.nlm.nih.gov/8632824/

Human Vision : high performing in recognition

Humans can tell a lot about a scene even from very little information.

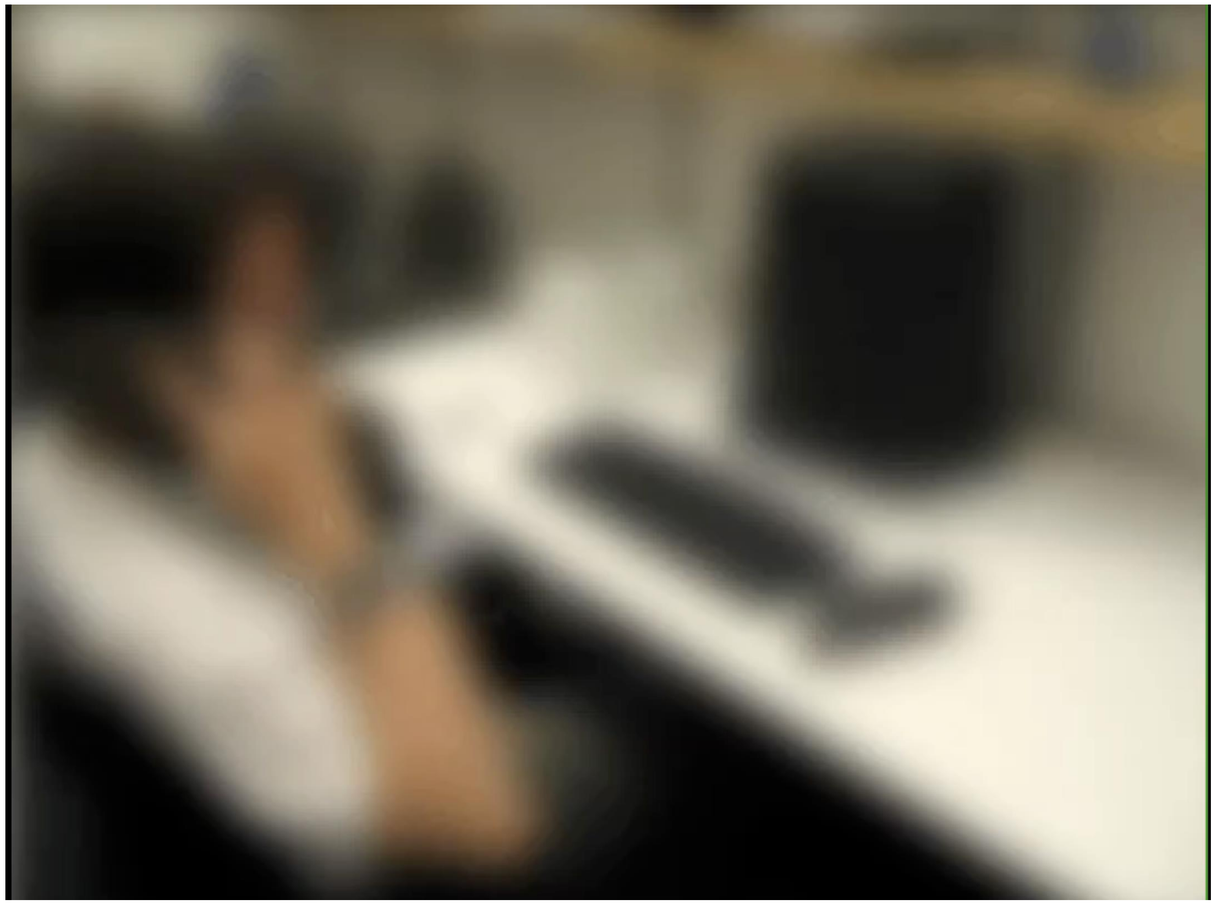


Source: “80 million tiny images” by Torralba, et al.

About this dataset: 80 million tiny images - Delete due to contentious images in 2020.
https://groups.csail.mit.edu/vision/TinyImages/

Human Vision : high performing in recognition

Importance of the context.



Human Vision : high performing in recognition

Importance of the context.

Computer vision

Input Image

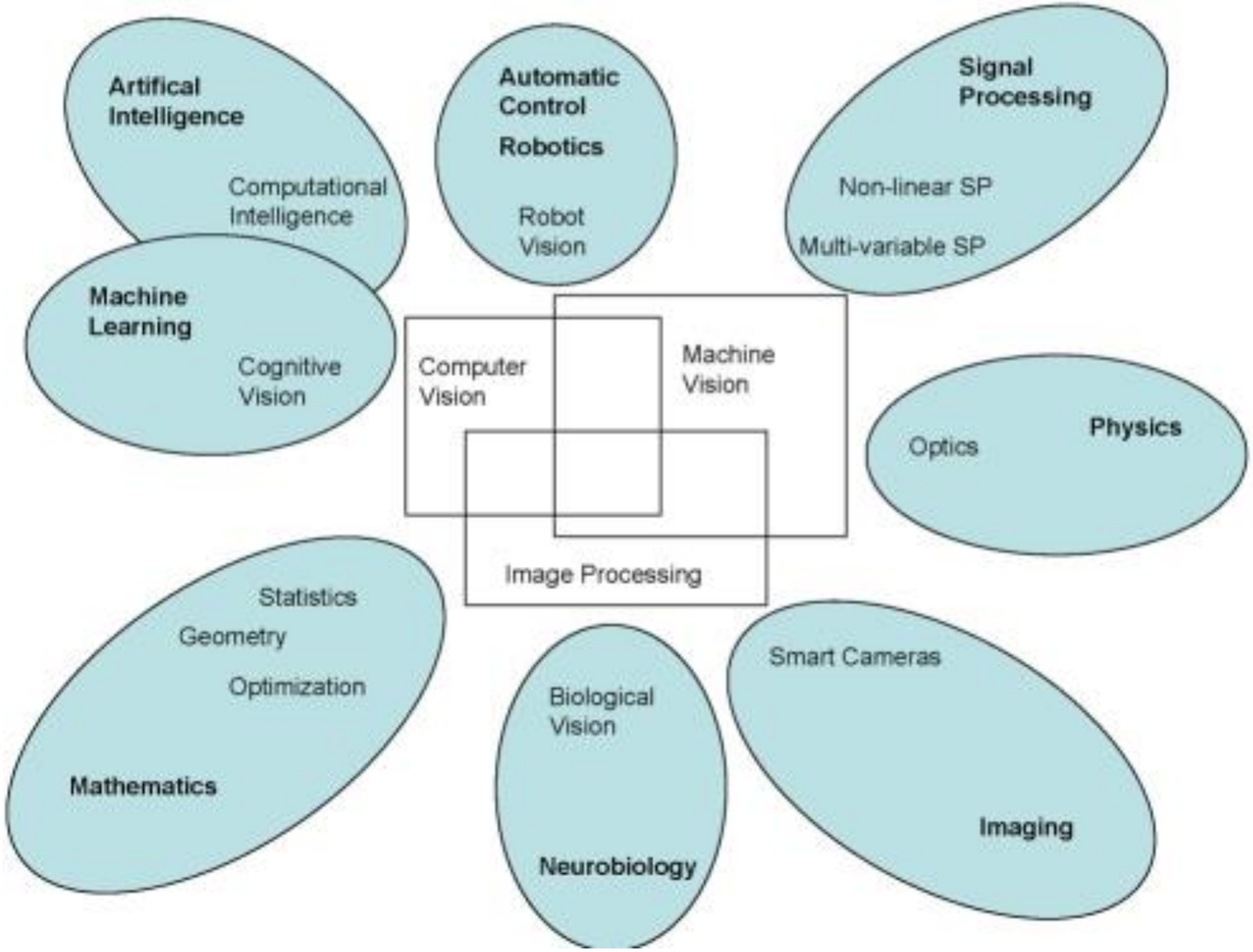
Sensing device

Interpreting device

Interpretations

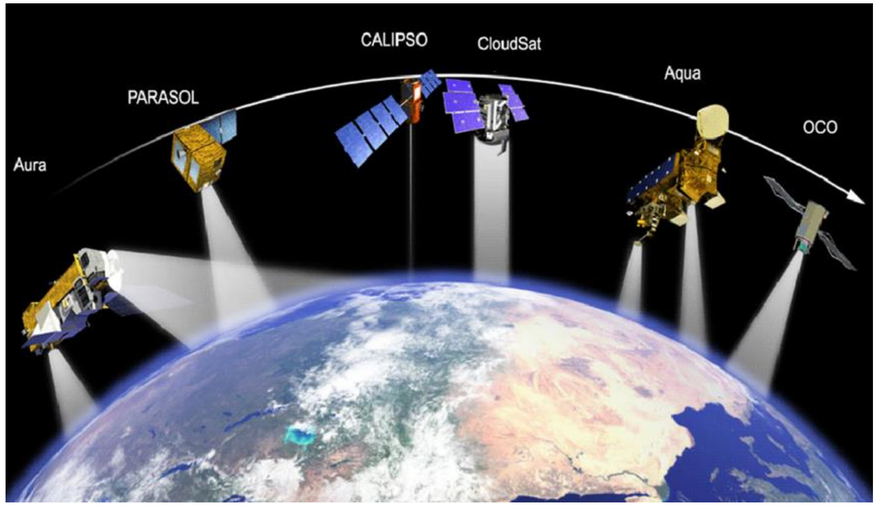
outdoor,
car,
roads,
bicycles,
building,
etc

Interdisciplinary

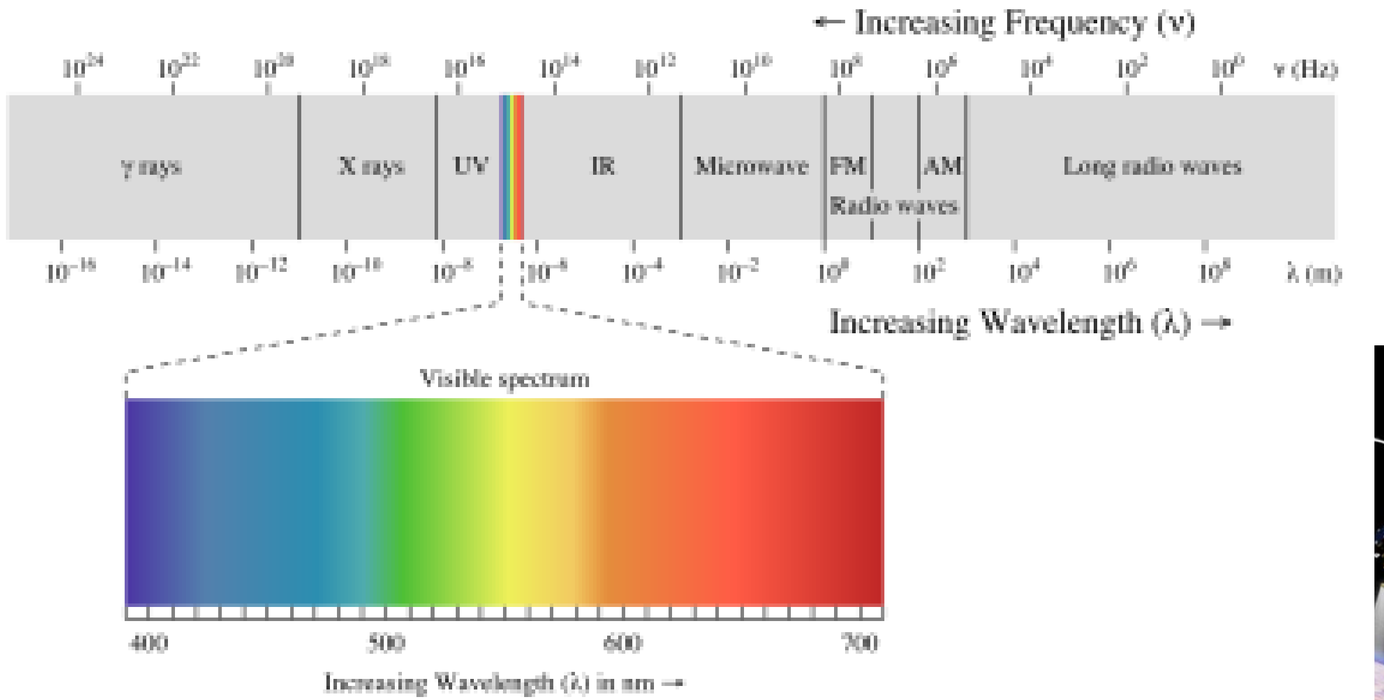


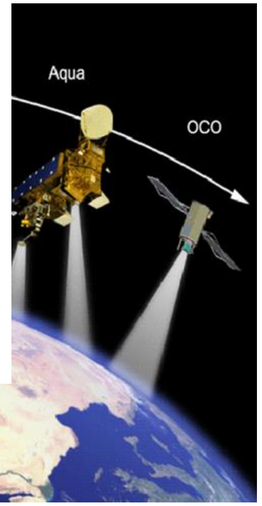
Why is it important?

- **Variety of applications**



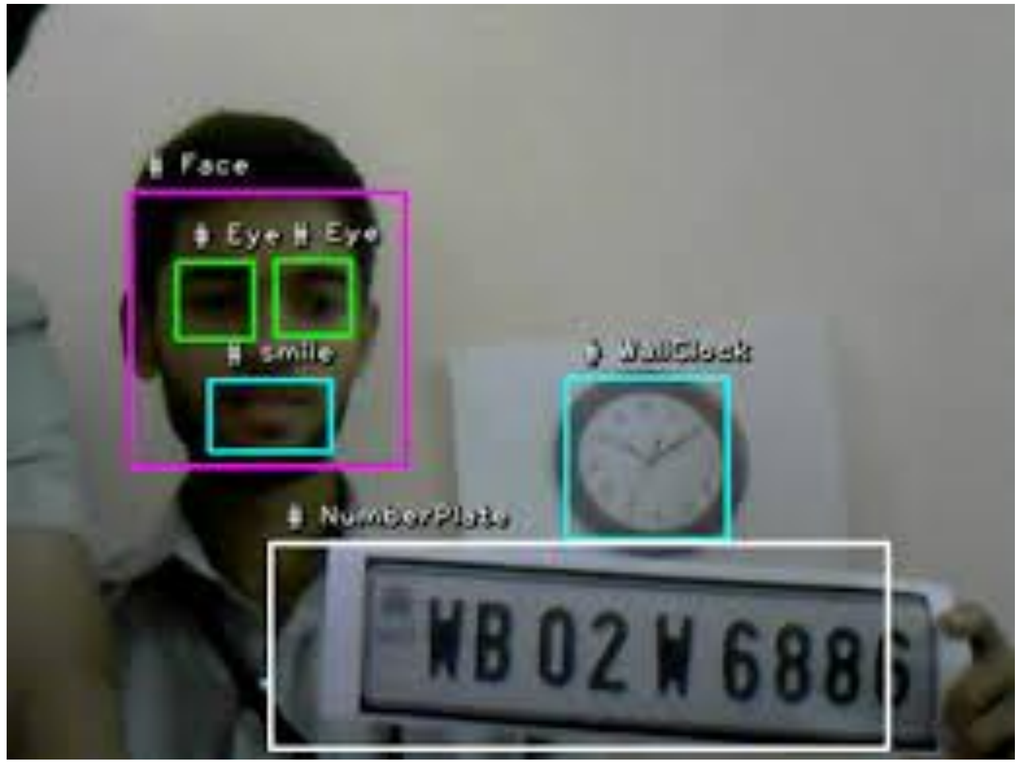
Why is it important?





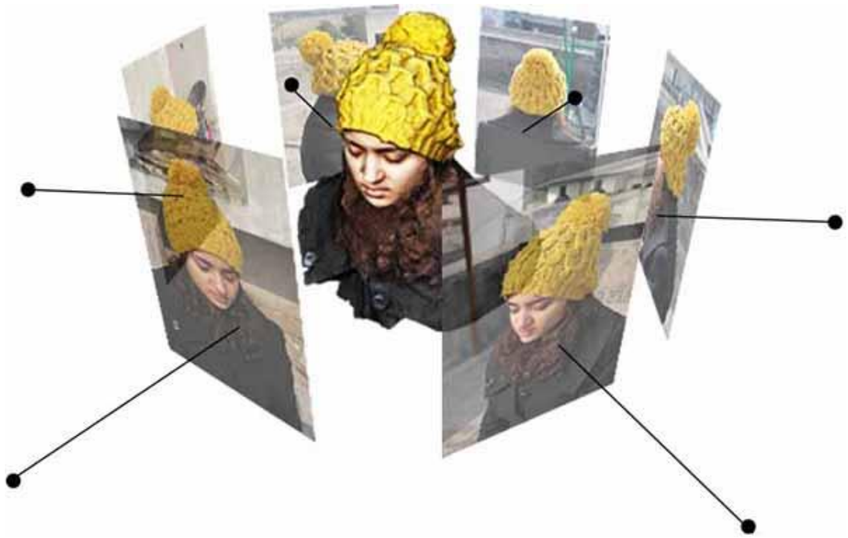
Where can it be applied?

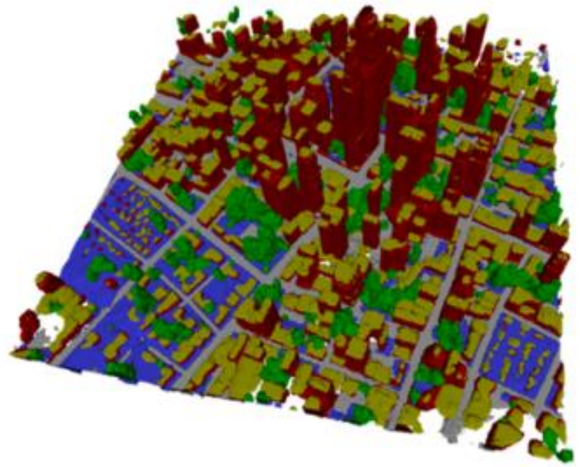
- **Object recognition, detection**



Where can it be applied?

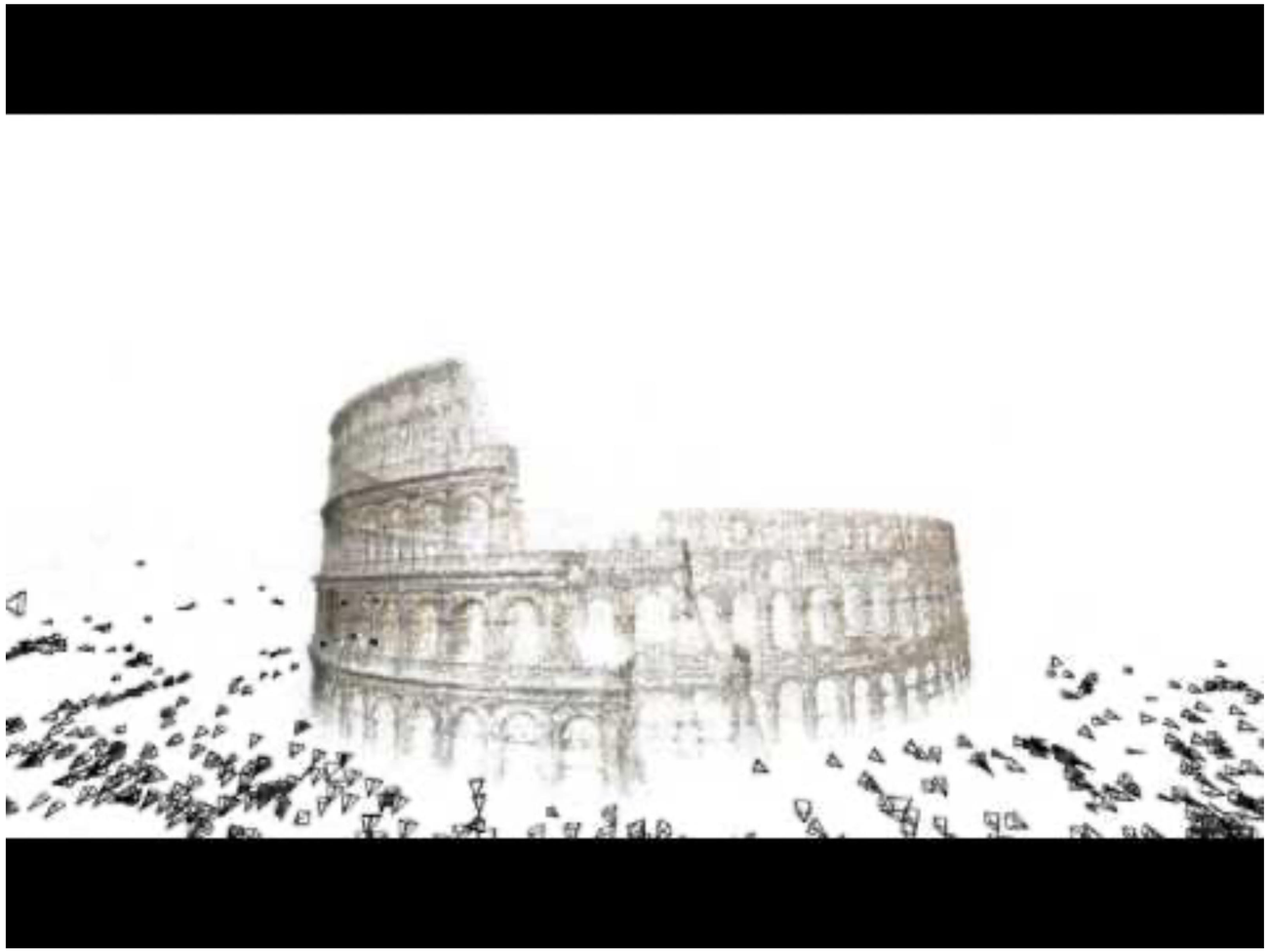
- **Object recognition, detection**
- **3D reconstruction**





Where can it be applied?

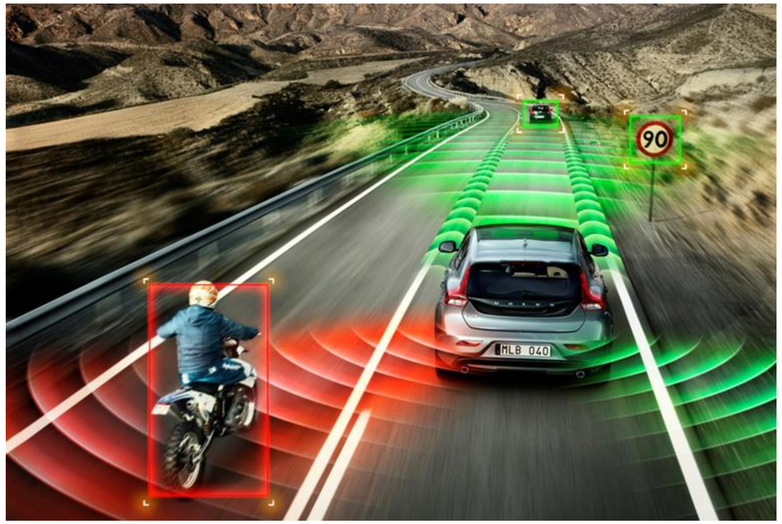
- **Object recognition, detection**
- **3D reconstruction :**
  - **building Rome in a day**

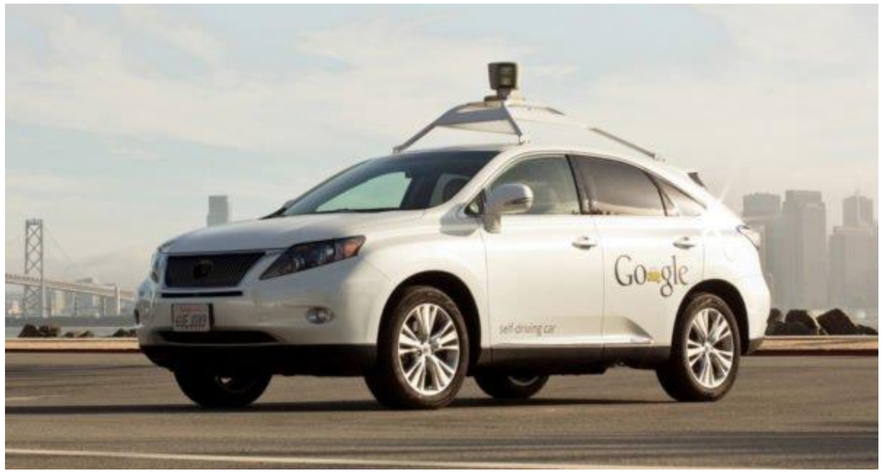


https://grail.cs.washington.edu/rome/
https://www.cs.cornell.edu/%7Esnavely/bundler/

Where can it be applied?

- **Object recognition, detection**
- **3D reconstruction**
- **Autonomous Cars**



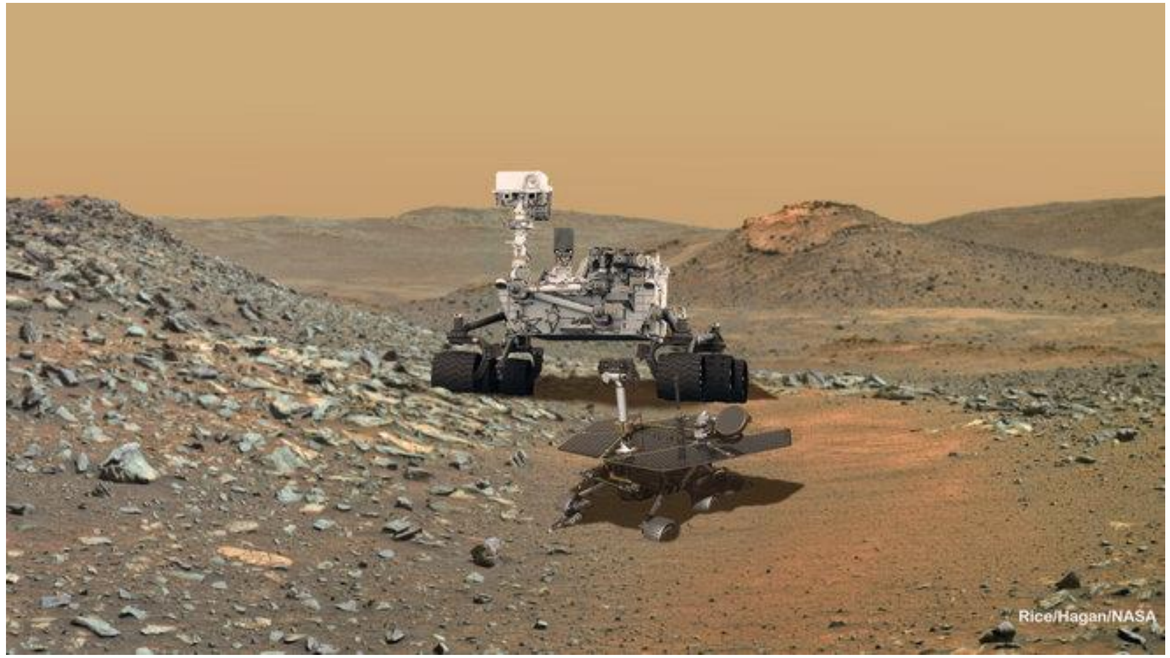


Where can it be applied?

- **Object recognition, detection**
- **3D reconstruction**
- **Autonomous Cars**
- **Games**

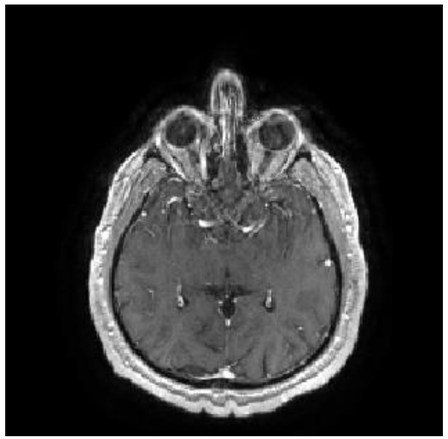
Where can it be applied?

- **Object recognition, detection**
- **3D reconstruction**
- **Autonomous Cars**
- **Games**
- **Earth and Space Observation**



Where can it be applied?

- **Object recognition, detection**
- **3D reconstruction**
- **Autonomous Cars**
- **Games**
- **Earth and Space Observation**
- **Medical Imaging**
- **….**

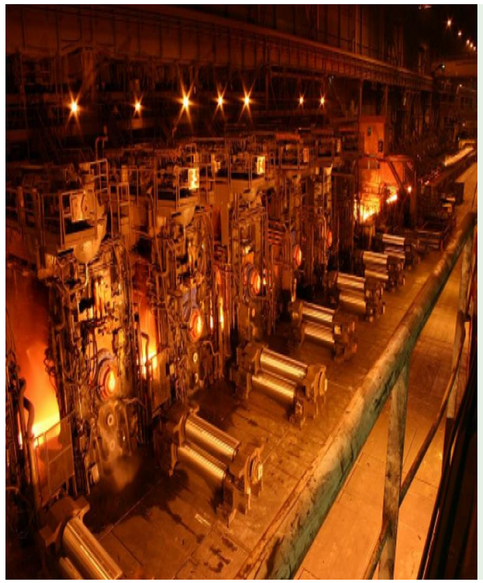


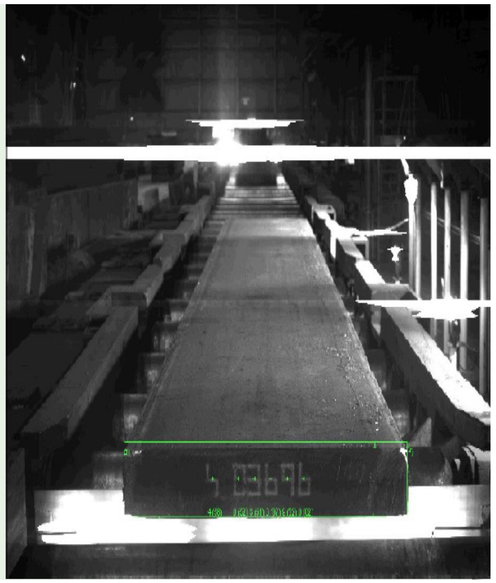
Where can it be applied?

- **Object recognition, detection**
- **3D reconstruction**
- **Autonomous Cars**
- **Games**
- **Earth and Space Observation**
- **Medical Imaging**
- **….**
- **Applications to our everyday life ...**

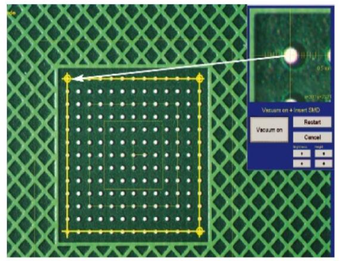
Where can it be applied?

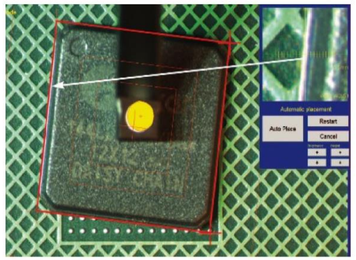
- **Production control and management**

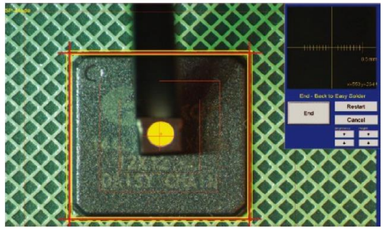




Control of steel plate production
Arcelor Mittal, Fos sur Mer

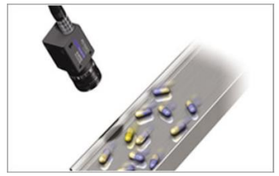






Placement of surface components

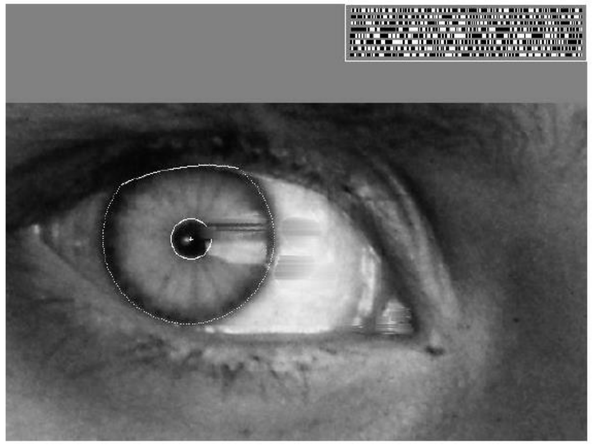
Pixfel©

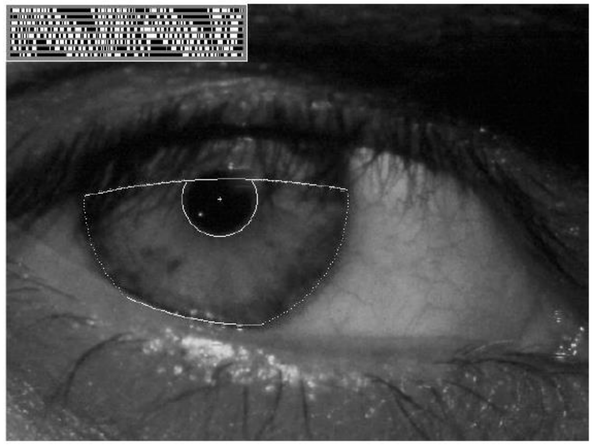


quality control compliant - non-compliant

Where can it be applied?

- **Security and Surveillance : biometry**





Source : Steve McCurry : `http://www.cl.cam.ac.uk/~jgd1000/afghan.html`

Where can it be applied?

- **Security and Surveillance : biometry**

Axis : Détection d'intrusion
Analyse de mouvement

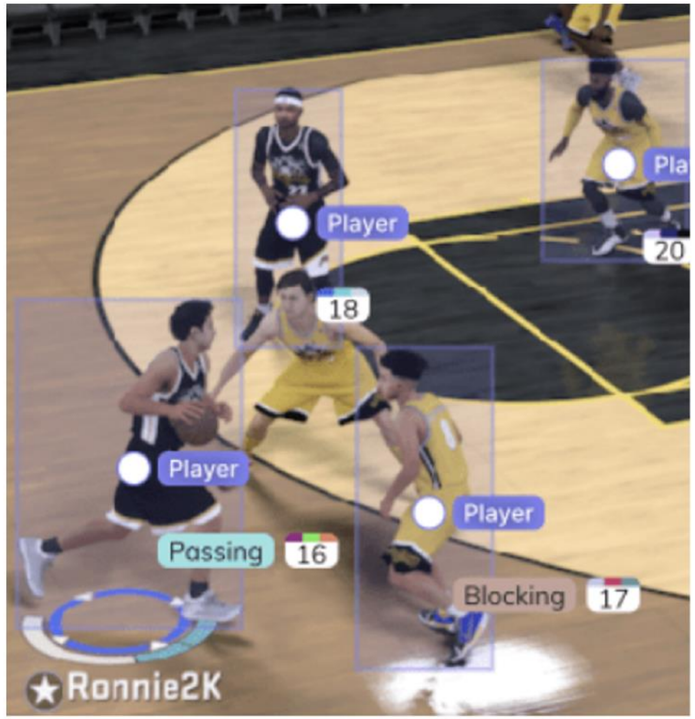
Poséidon, système d'IA pour la détection des noyades à l'aide de caméras aériennes et subaquatiques

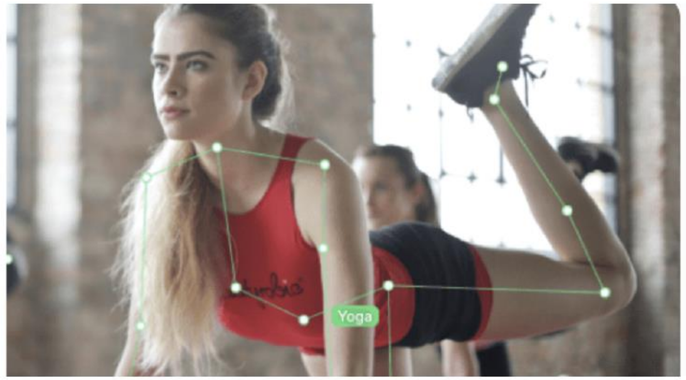
Poseidon: détecteur noyade

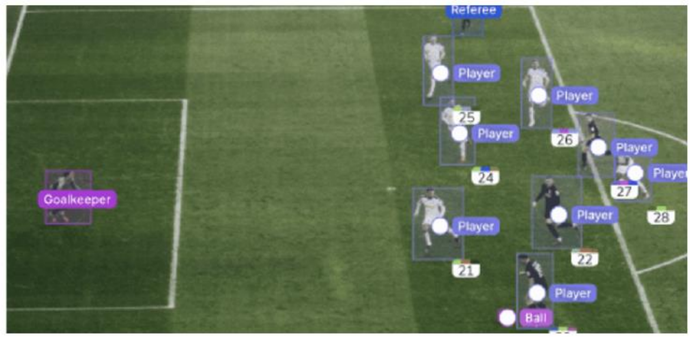
Axis : préservation de la vie privée

Where can it be applied?

- **In the sport domain**



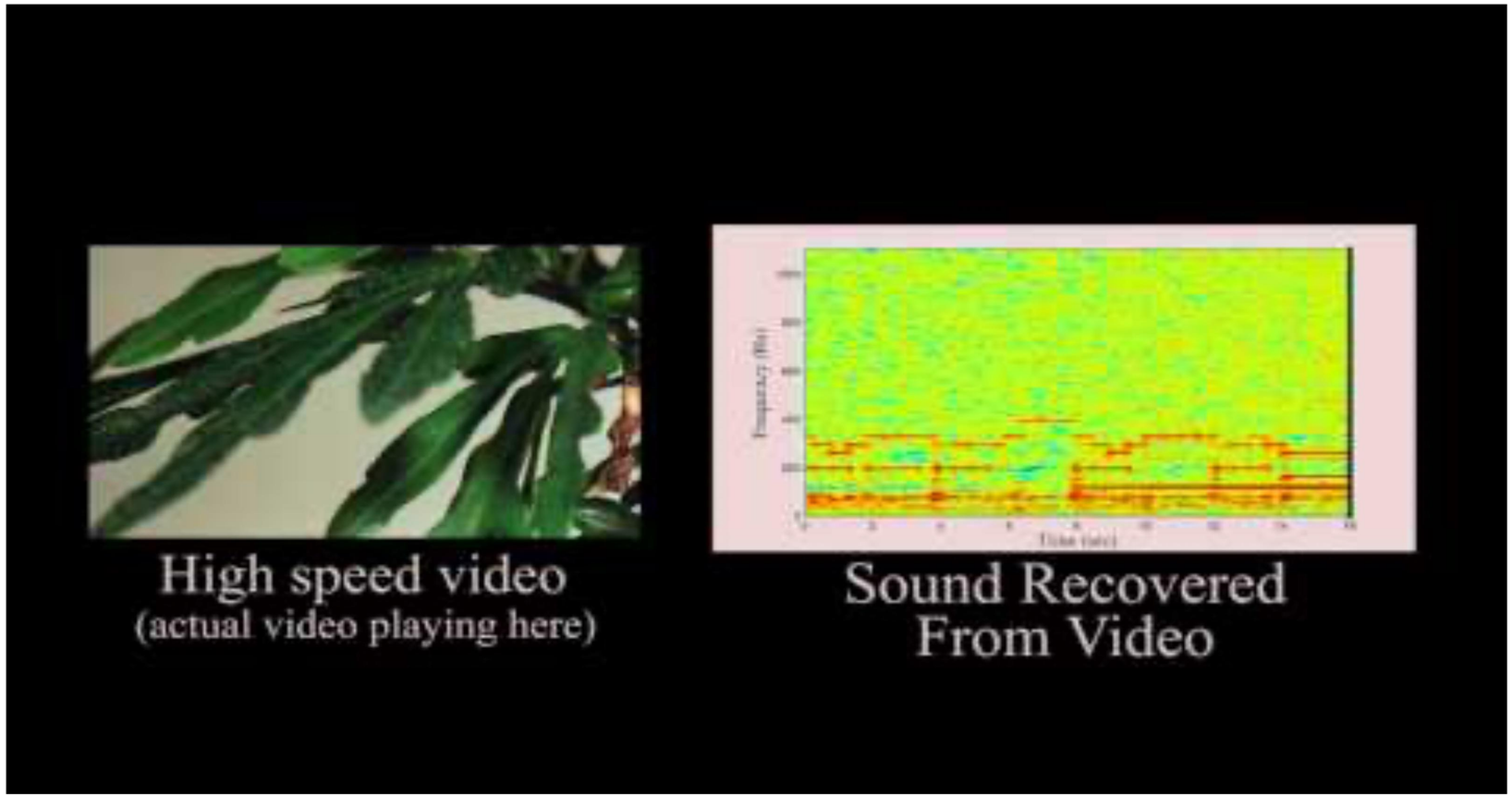




https://viso.ai/applications/visual-ai-in-sports/
https://www.v7labs.com/

Where can it be applied?

- **Another nice application : the visual microphone**

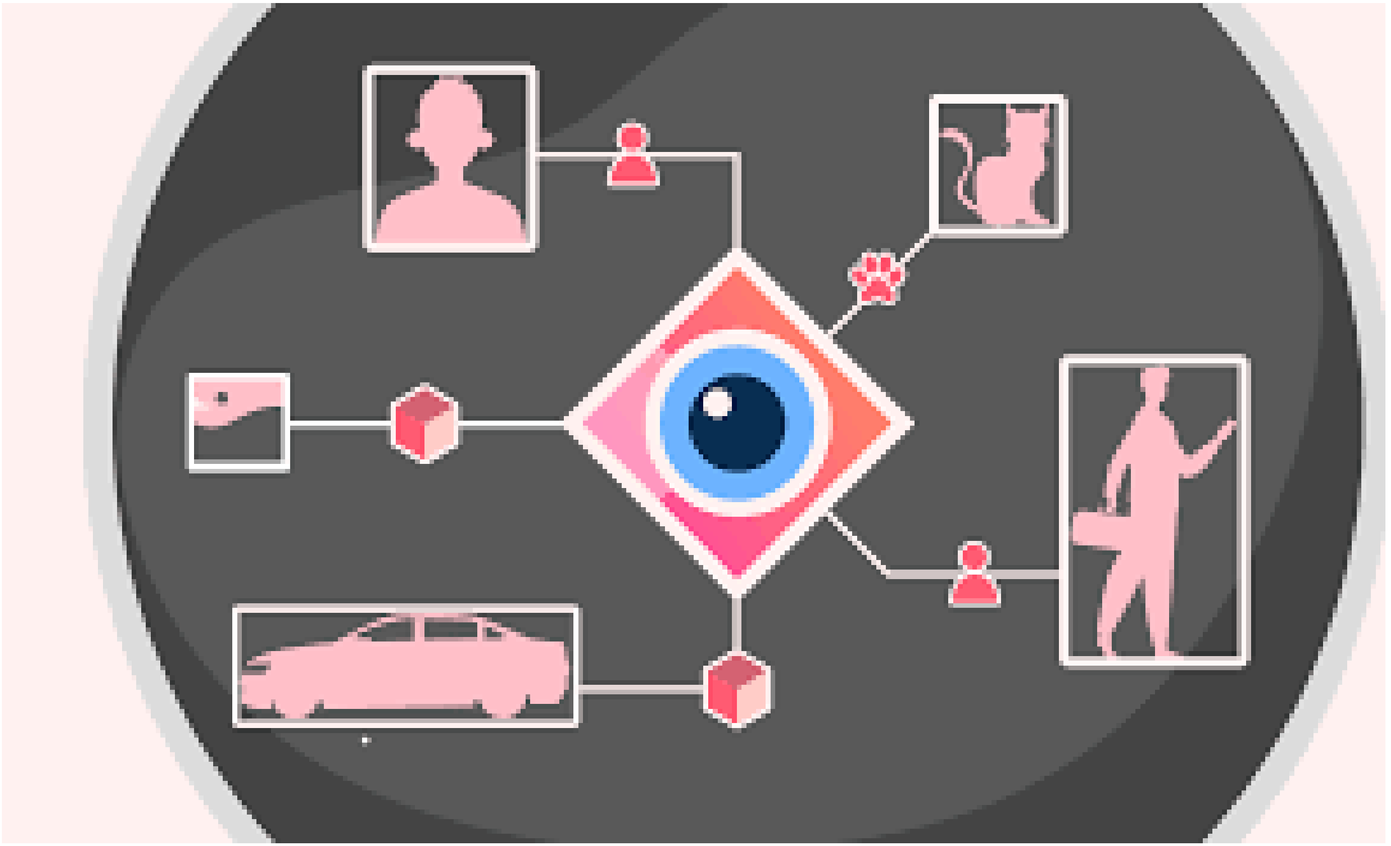


https://people.csail.mit.edu/mrub/VisualMic/

Computer Vision Topics?

- Low-level vision: feature, edge, texture, deblurring, visual saliency, …
- Mid-level vision: segmentation, superpixels, …
- High-level vision: object detection, object recognition, visual tracking, super resolution, image capturing, …
- Learning algorithms: Markov random field, conditional random field, graphical models, active learning, multi-view learning, ...

History of Computer Vision



Brief history of Computer Vision

In early 1970s, computer vision was viewed as the visual perception component of an ambitious agenda to mimic human intelligence and to endow robots with intelligent behaviour.

In 1966, Marvin Minsky asked Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw.”

Edge detection, stereo correspondences, line labeling, optical flow, ...

Brief history of Computer Vision

1970s

In early 1970s, compute erception component of an ambiti gence and to endow robots with intelli

In 1966, Marvin Minsky end the summer linking a camer mputer to describe what it saw."

Edge detection, stereo c cal flow, ...

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

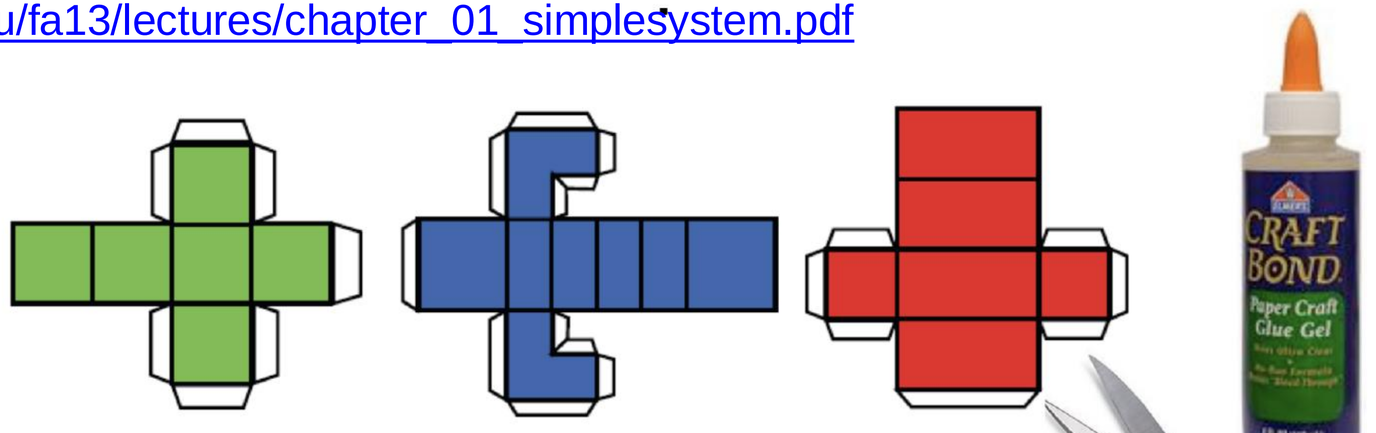
THE SUMMER VISION PROJECT

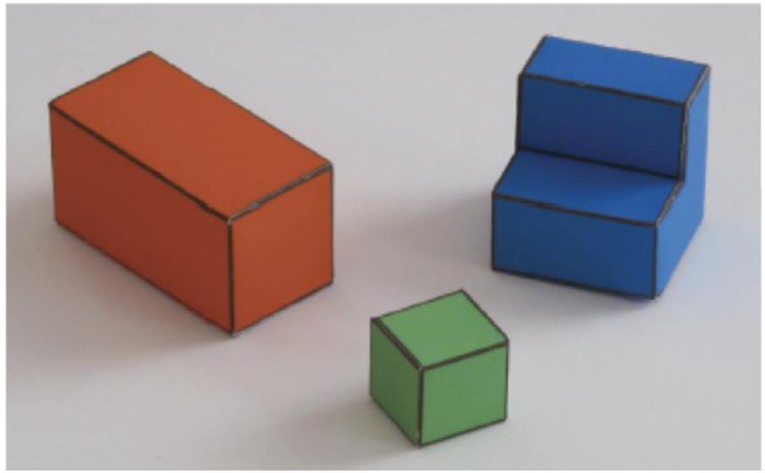
Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Brief history of Computer Vision

To go deeper on the Summer Vision Problem : Simple Vision System
http://6.869.csail.mit.edu/fa13/lectures/chapter_01_simplesystem.pdf





Brief history of Computer Vision

In the 1980s, a lot of attention was focused on more sophisticated mathematical techniques for performing quantitative image and scene analysis.

Methods using image pyramids, scale space processing, wavelets, better edge (e.g. canny) and contour algorithms, ...

Markov Random Field models, modeling and recognition ...

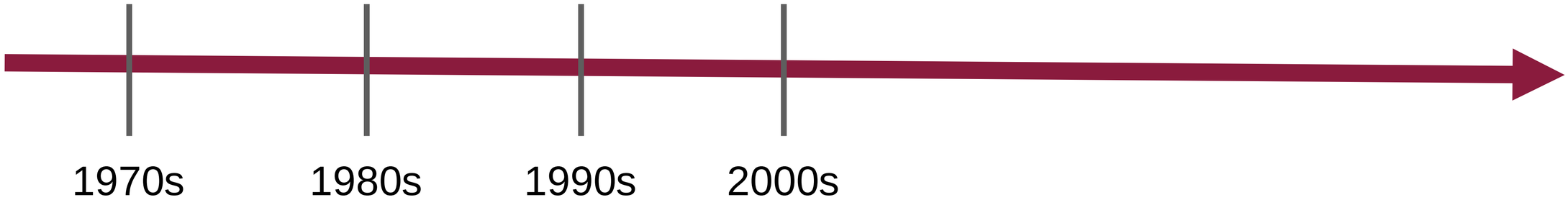
Brief history of Computer Vision

In the 1990s, all the previous but also some new topics had been explored.

Recognition, Projective reconstructions, 3D modeling, Physics-based vision, Optical Flow, stereo correspondences.

Tracking, image segmentation (mean shift), statistical learning (PCA), some interaction with computer graphics has established this period.

Brief history of Computer Vision

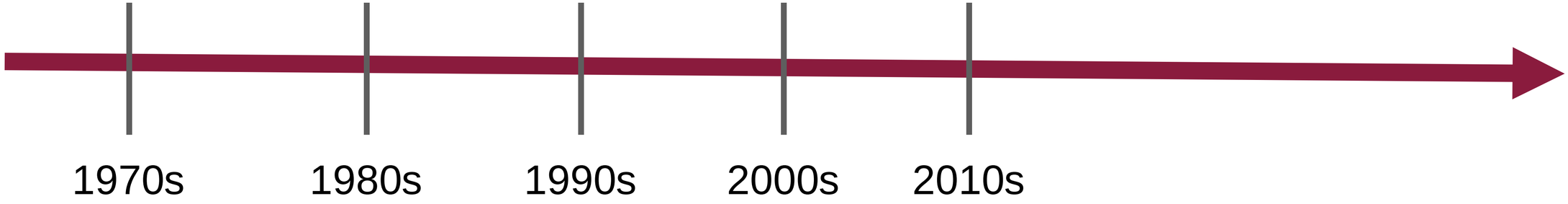


In the 2000s, all the previous but also some new topics had been explored.

Computational photography, Texture synthesis, Object recognition and feature-based techniques (SIFT, …), Scene classification, ...

Efficient optimization methods had been proposed (message passing, loopy belief propagation)

Brief history of Computer Vision

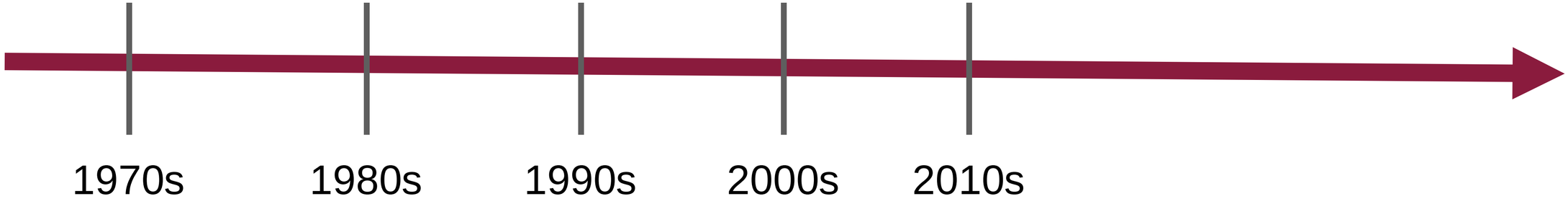


In the 2010s dominates a lot of the visual recognition research in our community, is the application of sophisticated machine learning techniques to computer vision problems.

Currently we have a lot of data ….

Deep Learning (Unsupervised, Supervised)

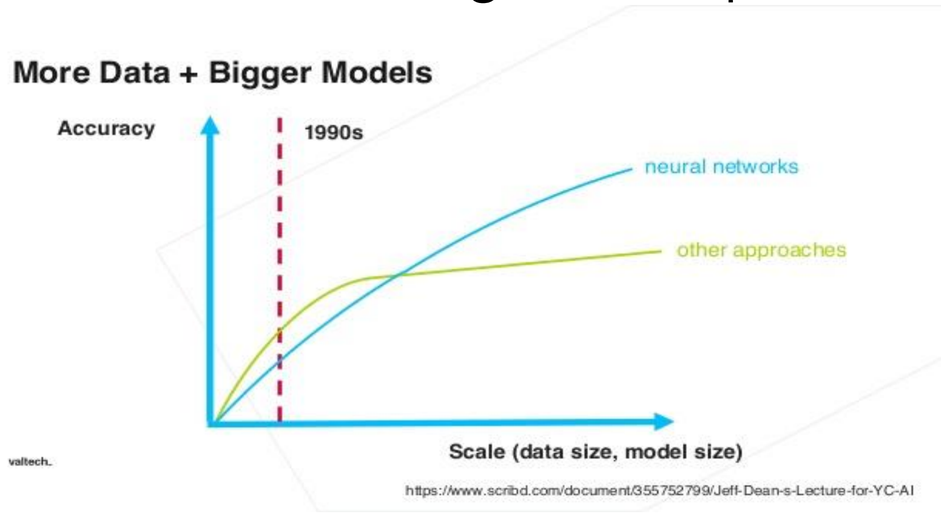
Brief history of Computer Vision



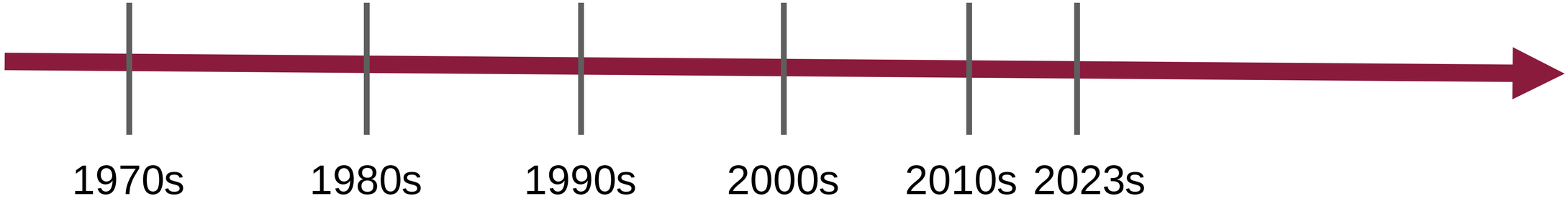
In the 2010s dominates a lot of the visual recognition research in our community, is the application of sophisticated machine learning techniques to computer vision problems.

Currently we have a lot of data ….

Deep Learning (Unsupervised, Supervised)

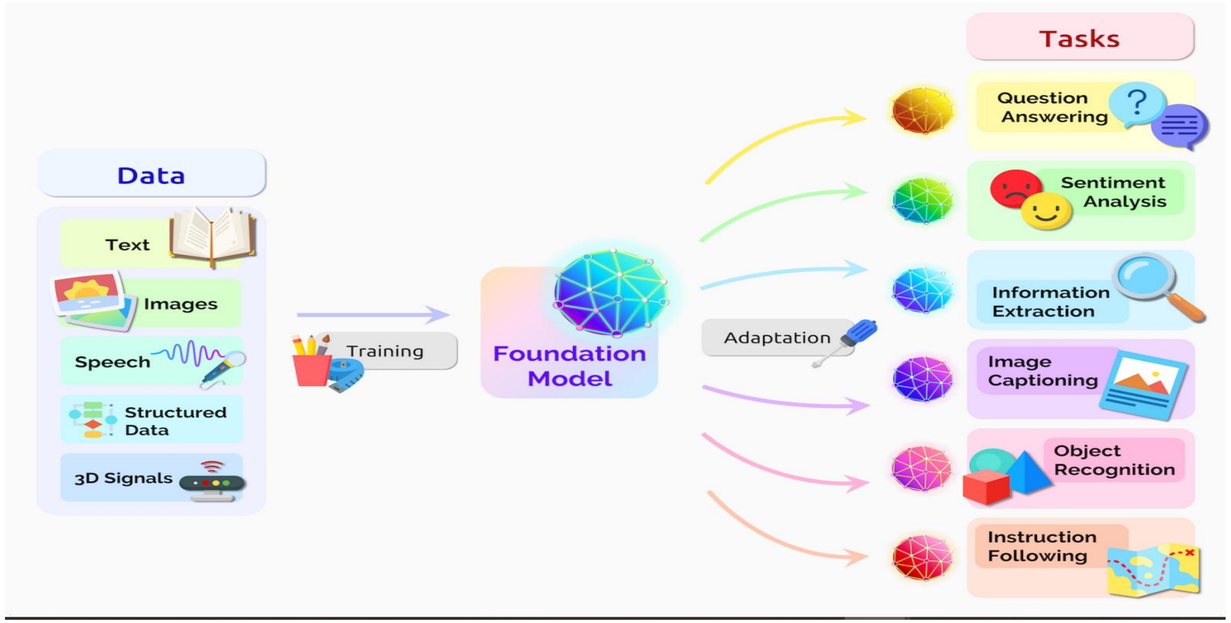


Brief history of Computer Vision



In recent years, we have the apparition of **foundation models** : A new major paradigm for building AI system.

Train a model on broad data and adapt it to a wide range of downstrams tasks.



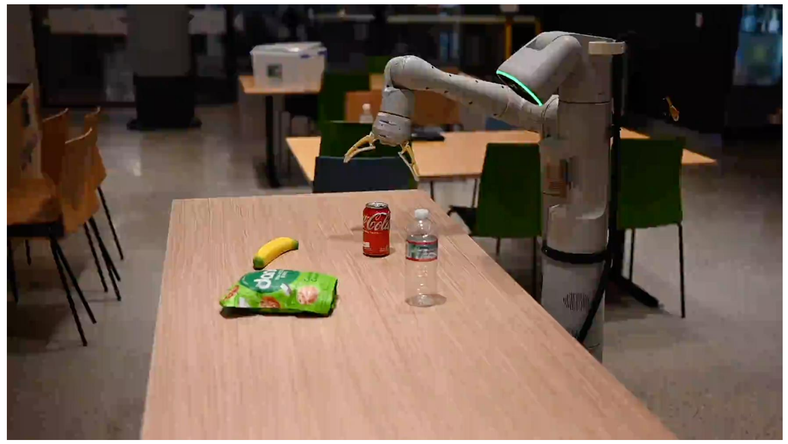
Bommasani et al., "**On the Opportunities and Risks of Foundation Models**"

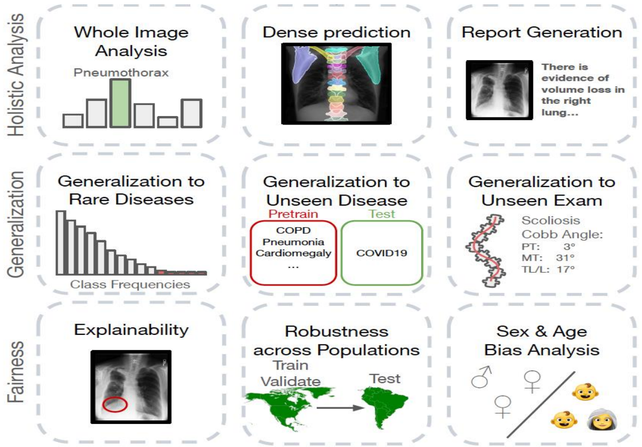
Brief history of Computer Vision

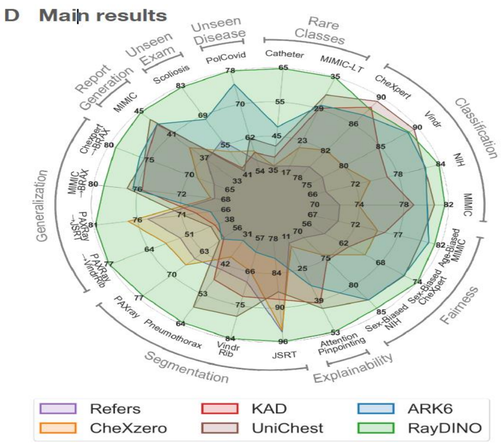
The Foundation model era

Segment Anything by Meta AI

RT-2: Vision-Language-Action Models by Google Deep Mind





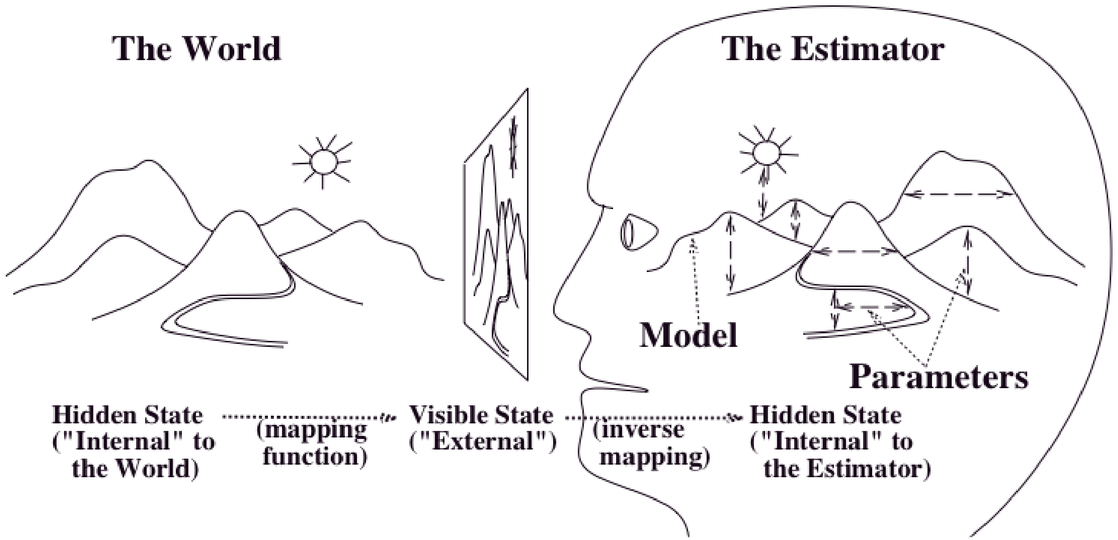


Moutakanni et al.,**Xray Dino**

Approaches of Computer Vision

Modeling + Algorithms

- Build a simple model of the world.
- Find some probably good algorithms.
- Experiment on real world.
- Update model.
  - *Problem:* Too often the models are simplistic or intractable



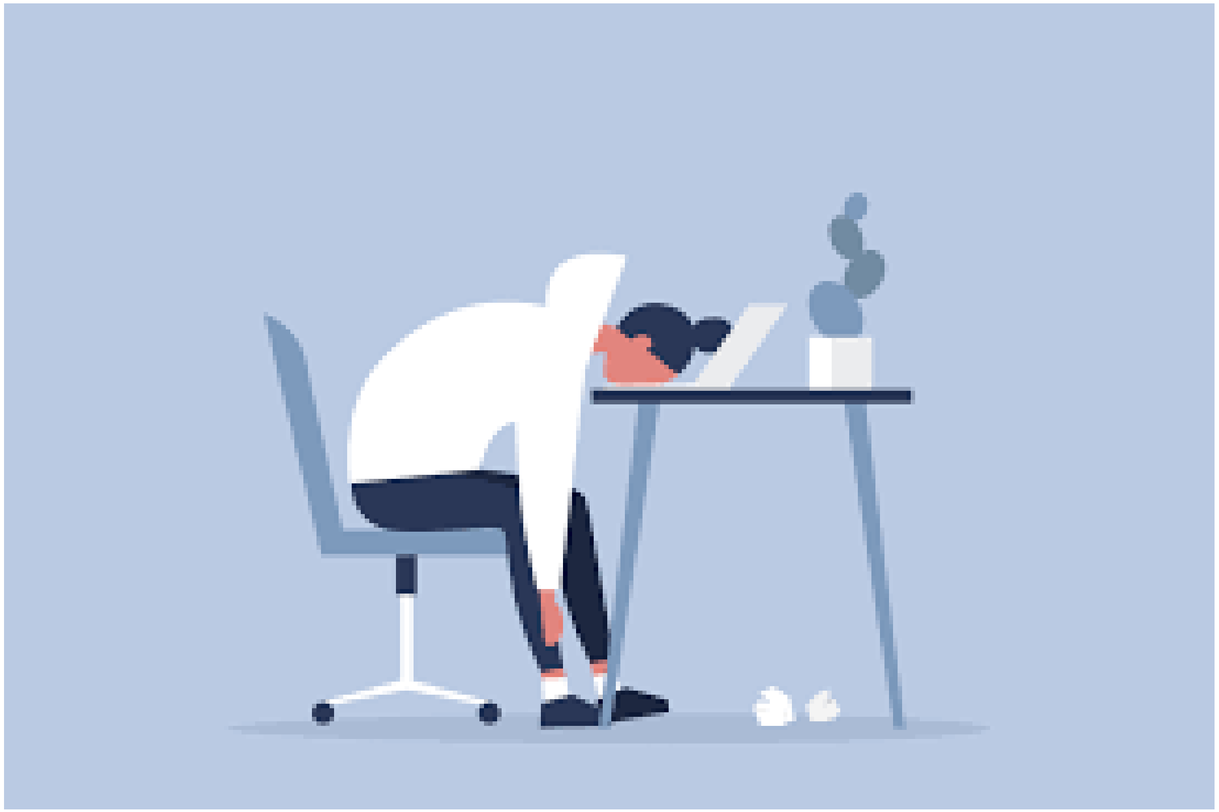
Bayesian inference

- Bayes law: $P(A|B) = [P(B|A) \cdot P(A)]/P(B)$
- $P(\text{world}|\text{image}) = [P(\text{image}|\text{world}) \cdot P(\text{world})]/P(\text{image})$
- $P(\text{image}|\text{world})$ can be modeled
  - Geometry of projection
  - Physics of light and reflection
- $P(\text{world})$ means modeling objects in the world.
- Leads to statistical/ learning approaches.
  - *Problem:* Too often probabilities can't be known and are invented.

Engineering

- Focus on definite tasks with clear requirements.
- Try ideas based on theory and get experience about what works.
- Try to build reusable modules.
  - *Problem:* Solutions that work under specific conditions may not generalize

Why Computer Vision is difficult?



Challenges 1: view point variation

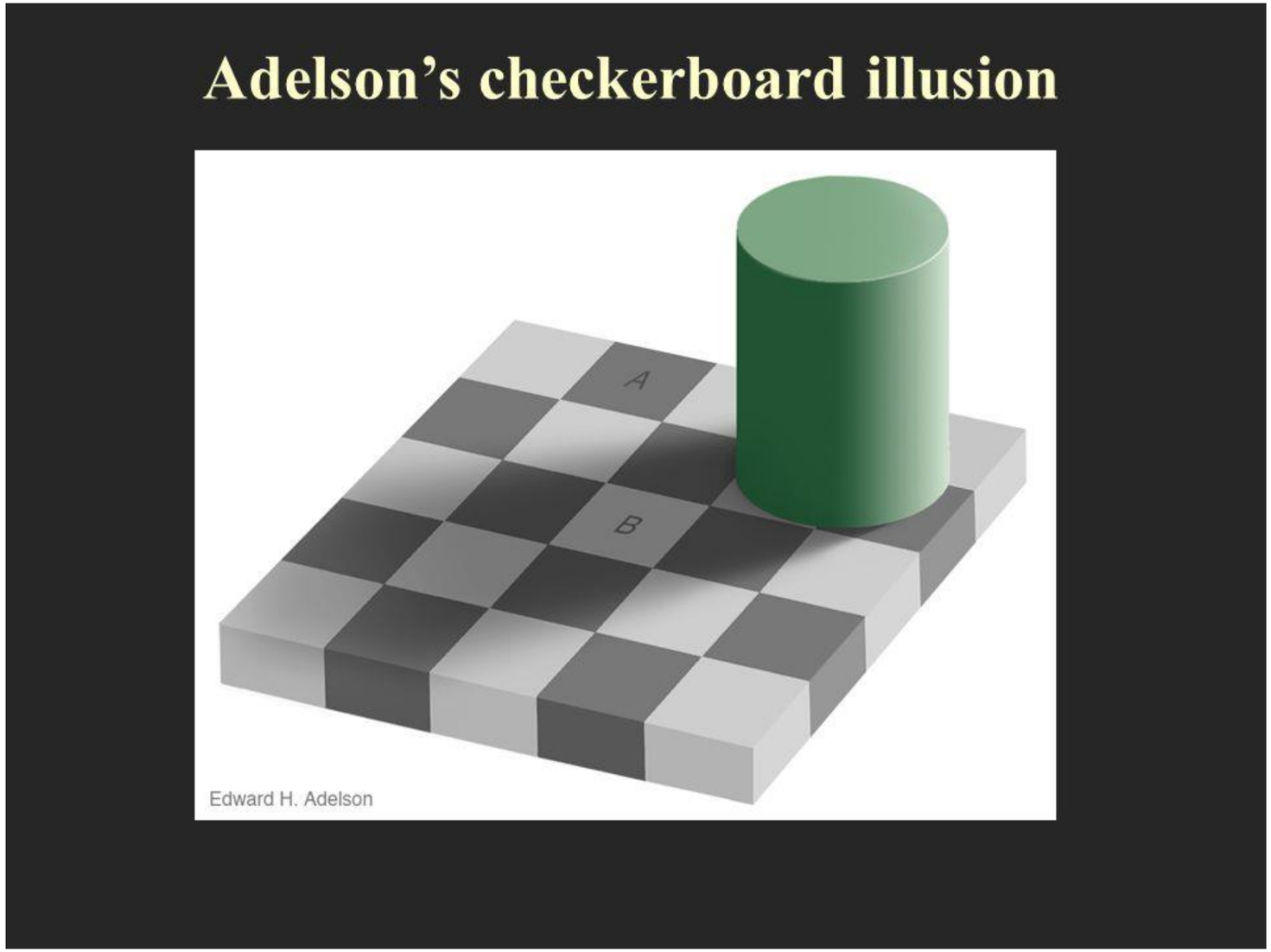
Michelangelo 1475-1564

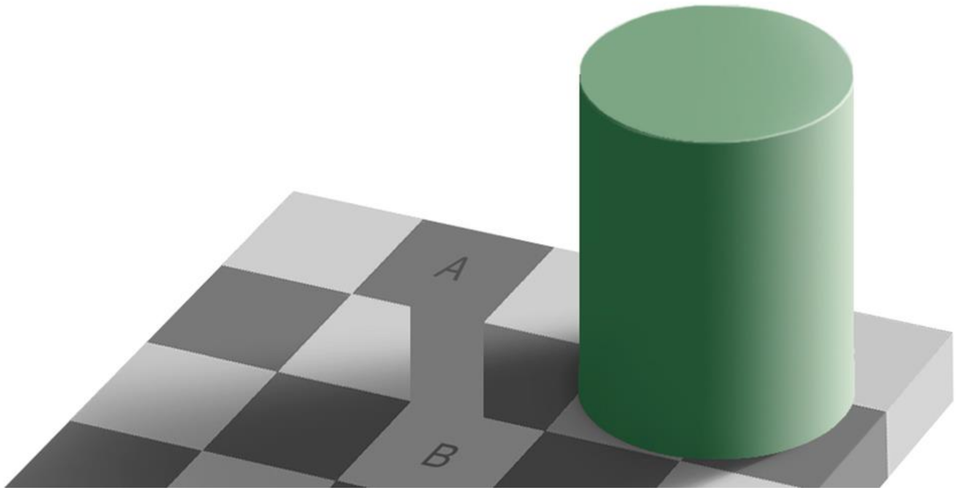
slide credit: Fei Fei, Fergus & Torralba

Challenges 2: illumination

slide credit: S. Ullman

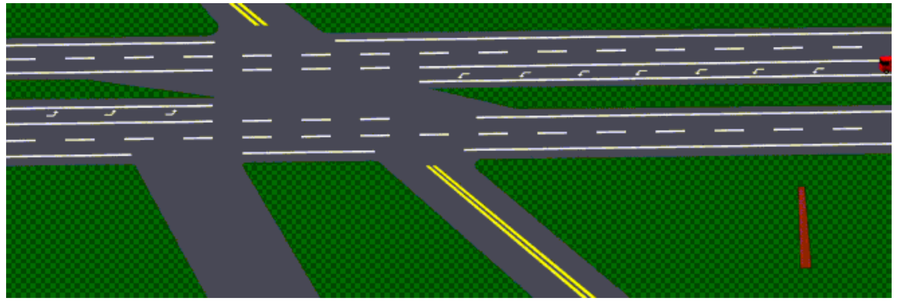
Challenges 2: illumination





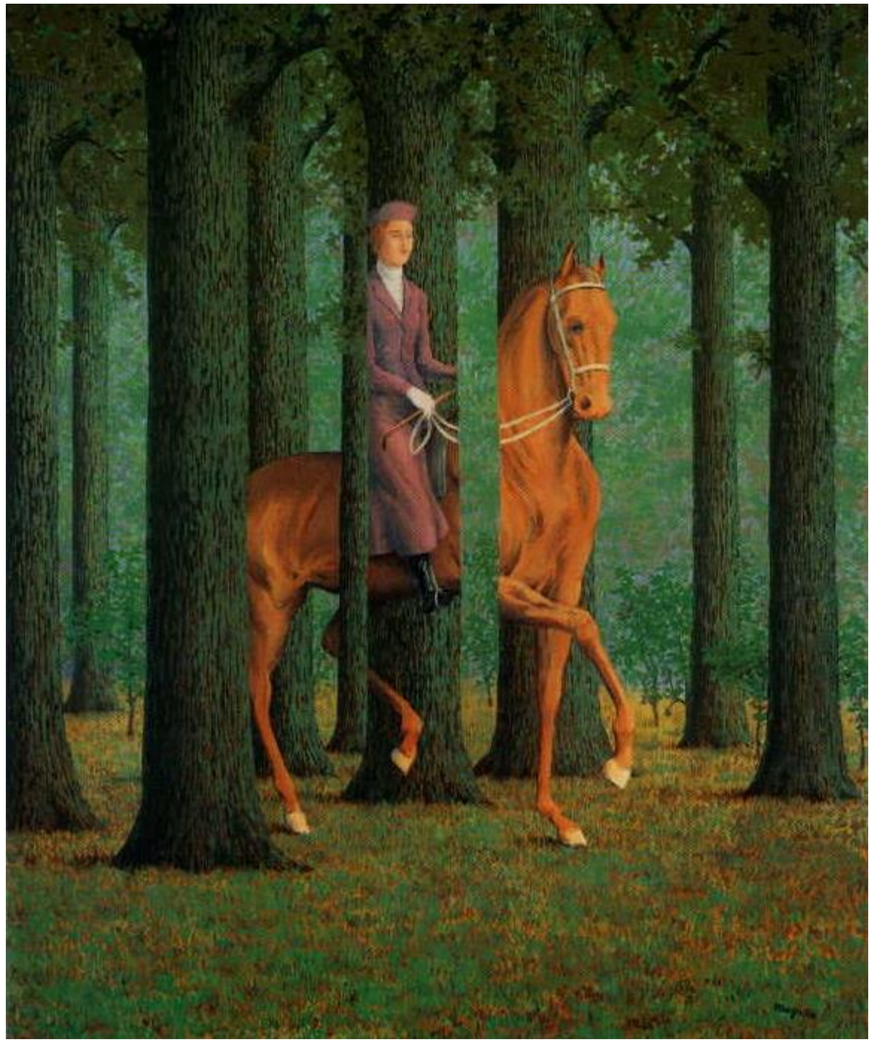
Challenges 2: illumination

NHTSA: Neither Autopilot nor the driver noticed the white side of the tractor-trailer against a brightly lit sky, so the brake was not applied.



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

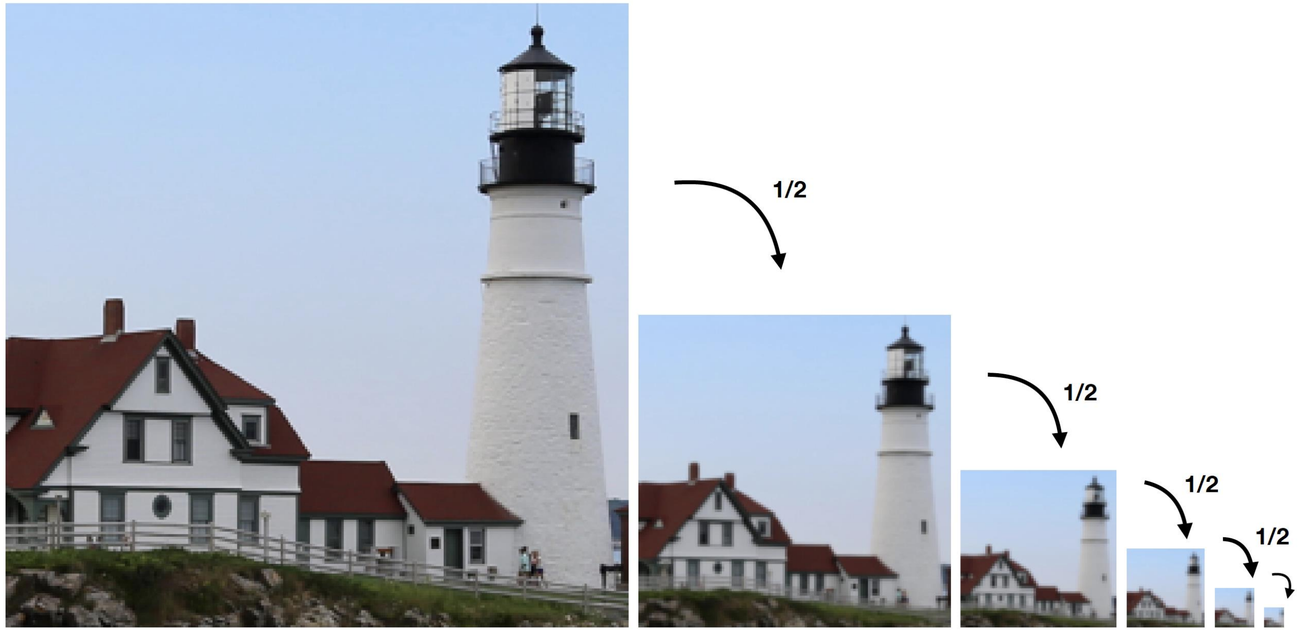
Challenges 3: occlusion



Magritte, 1957

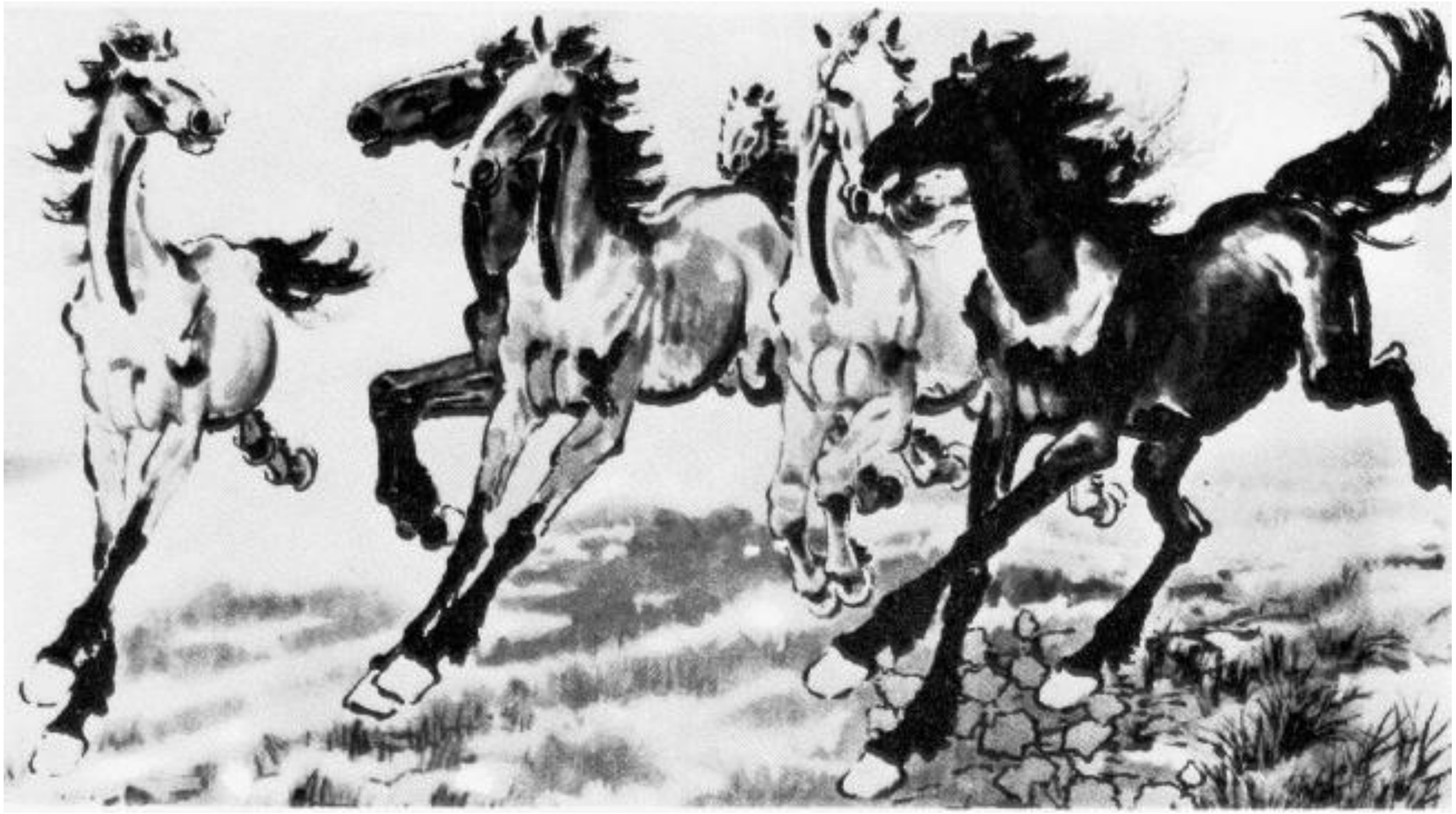
slide credit: Fei Fei, Fergus & Torralba

Challenges 4: scale



slide credit: Fei Fei, Fergus & Torralba

Challenges 5: deformations

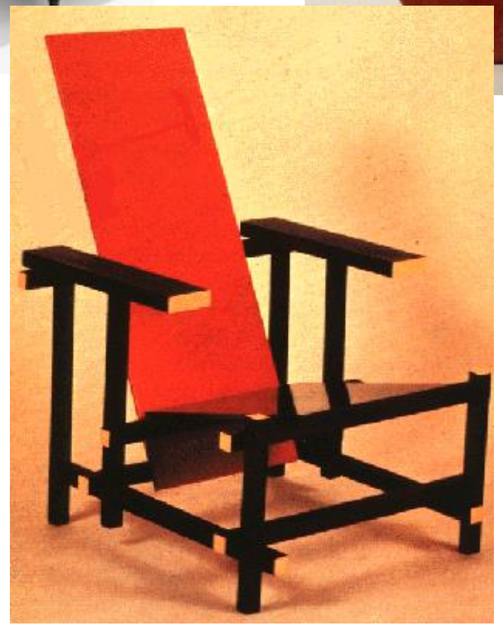


Xu, Beihong 1943

slide credit: Fei Fei, Fergus & Torralba

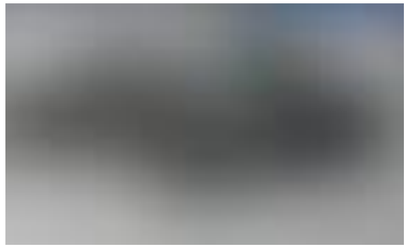
Challenges 6: background clutter

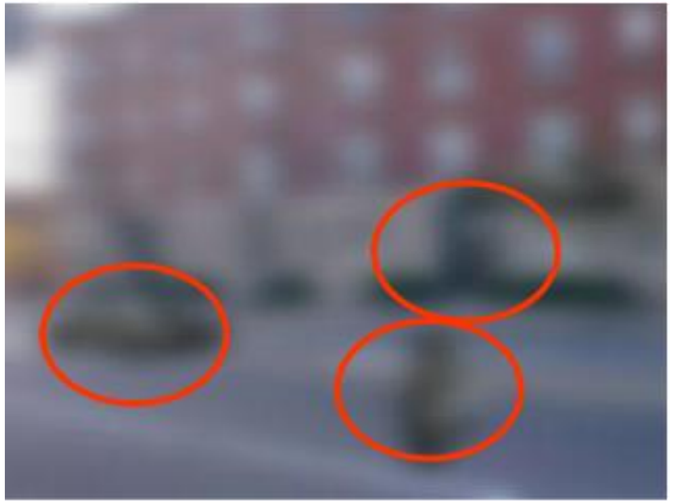
Challenges 7: object intra-class variation



slide credit: Fei Fei, Fergus & Torralba

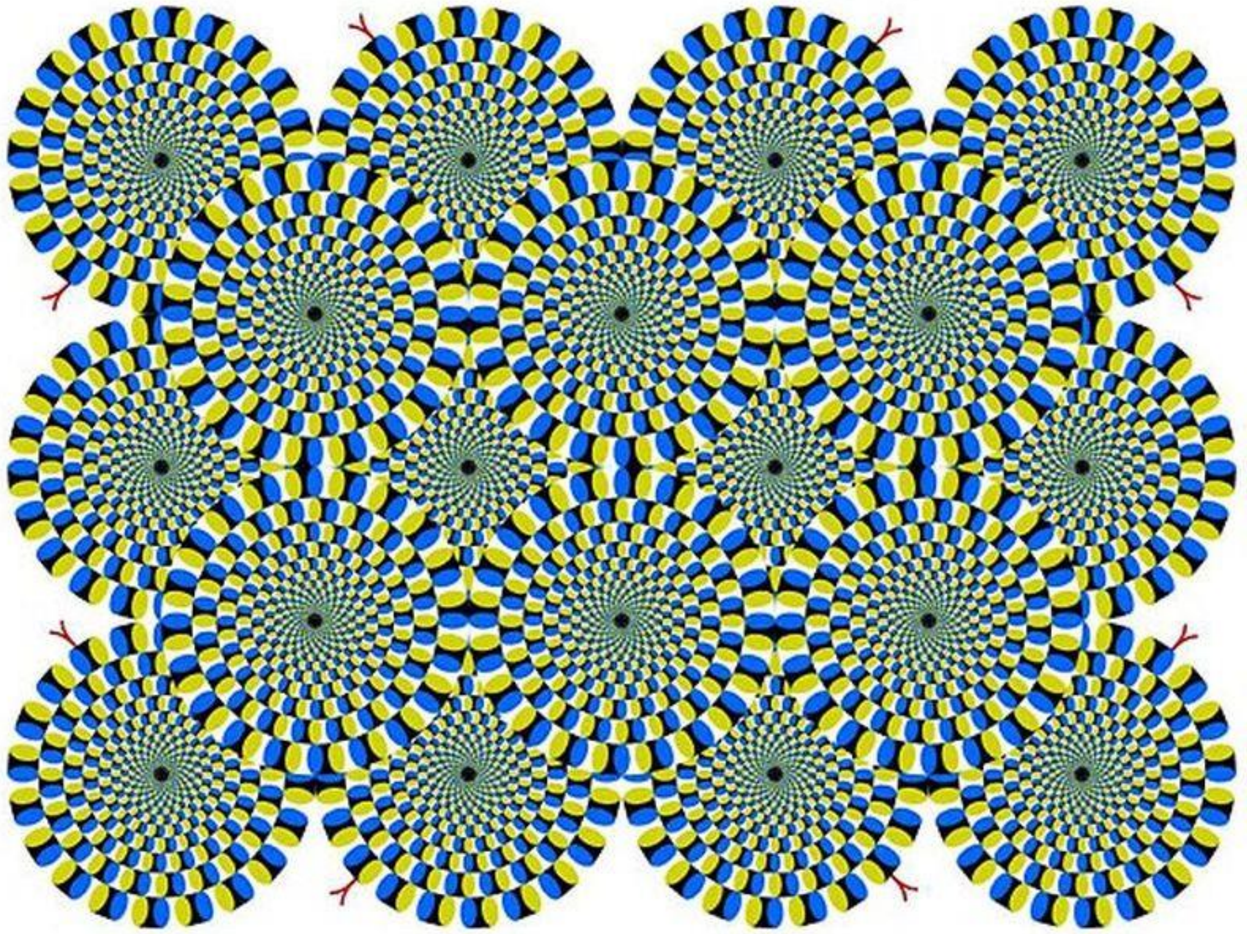
Challenges 8: local ambiguity





slide credit: Fei Fei, Fergus & Torralba

Challenges 9: illusions/ motions



Challenges 10: the world behind the image

I can't really describe the picture but I do see indoor, table, room.

We need prior knowledge to make meaningful interpretations of an image.

Challenges 11: the world behind the image

Generated Images for the caption: “A salmon swimming.”

Even for the recent AI models Vision is quite difficult

Challenges or opportunities?

- Images are confusing, but they also reveal the structure of the world through numerous clues.
- Our job is to interpret the clues!

Bottom line

- Perception is an inherently ambiguous problem
  - Many different 3D scenes could have given rise to a particular 2D picture

- Possible solutions
  - Bring in more constraints (or more images)
  - Use prior knowledge about the structure of the world.
- Need both exact measurements and statistical inference!

Image Sensing Pipeline

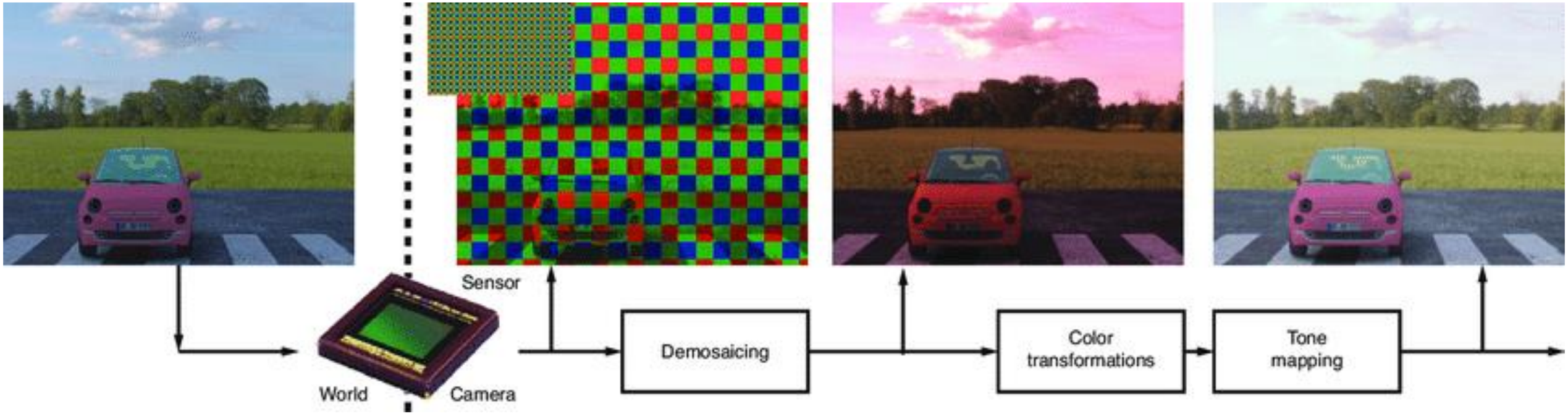
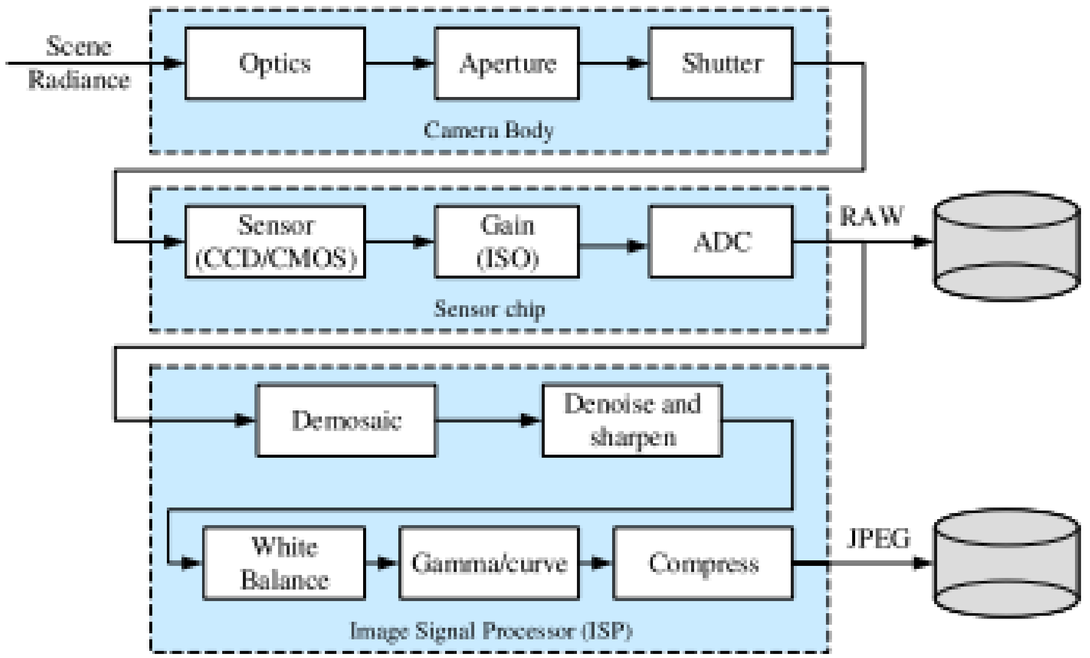


Image Sensing Pipeline

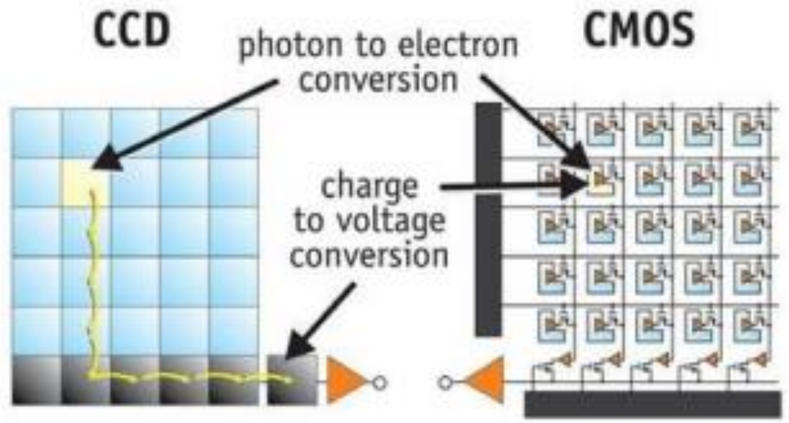


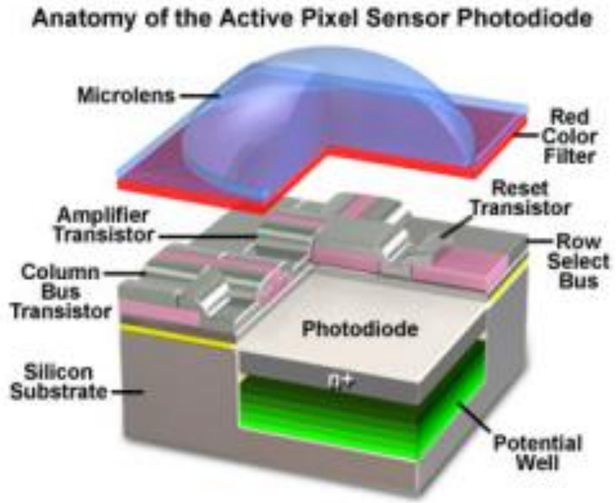
- The image sensing pipeline can be divided into three stages:
  - **Physical light transport** in the camera lens/ body
  - **Photon measurement** and conversion on the sensor chip
  - **Image signal processing** (ISP) and image compression

Shutter

- A focal plane shutter is positioned just in front the image sensor/ film
- Most digital cameras use a combination of mechanical and electronic shutter
- The shutter speed (exposure time) controls how much light reaches the sensor
- It determines if an image appears over-/underexposed, blurred or noisy

Sensor





- CCDs move charge from pixel to pixel and convert it to voltage at the output node
- CMOS images convert charge to voltage inside each pixel and are standard
- Larger chips (full frame = 35mm) are more photo-sensitive => less noise

https://meroli.web.cern.ch/lecture_cmos_vs_ccd_pixel_sensor.html

Color Filter Arrays

G	R	G	R
B	G	B	G
G	R	G	R
B	G	B	G

Bayer RGB Pattern

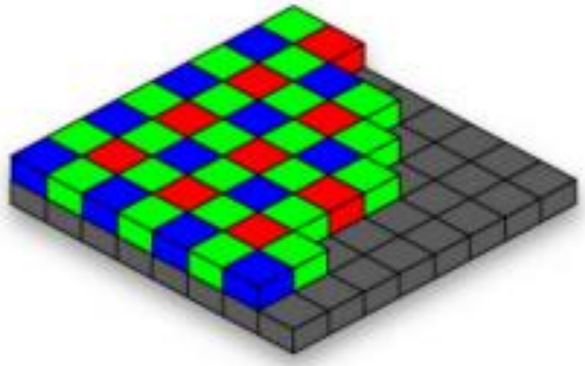
rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb
rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb

Interpolated Pixels

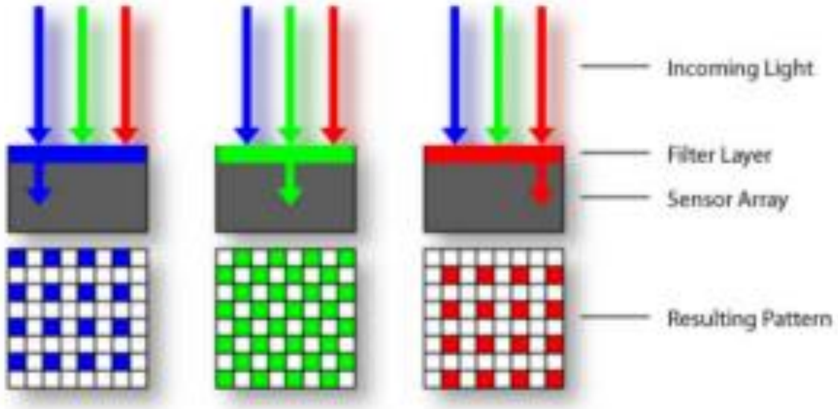
- To measure color, pixels are arranged in a **color array**, e.g.: Bayer RGB pattern
- Missing colors at each pixel are interpolated from the neighbors (demosaicing)

Color Filter Arrays

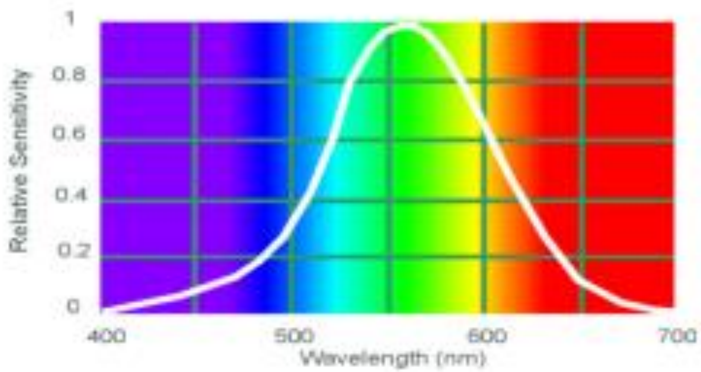
Bayer grid



Estimate missing components from neighboring values (demosaicing)

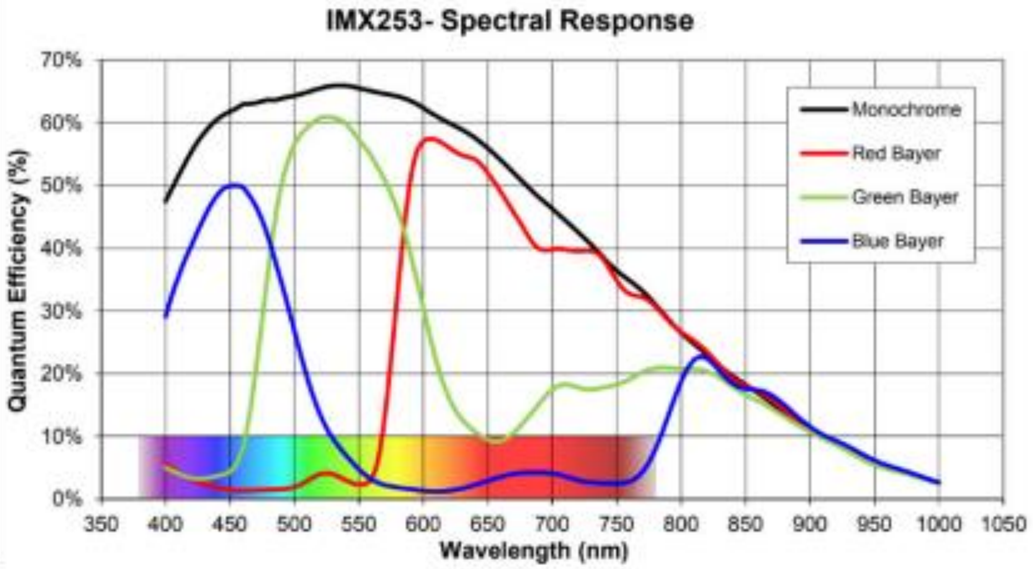


Why more green?



Slide Credits: Steve Seitz

Color Filter Arrays

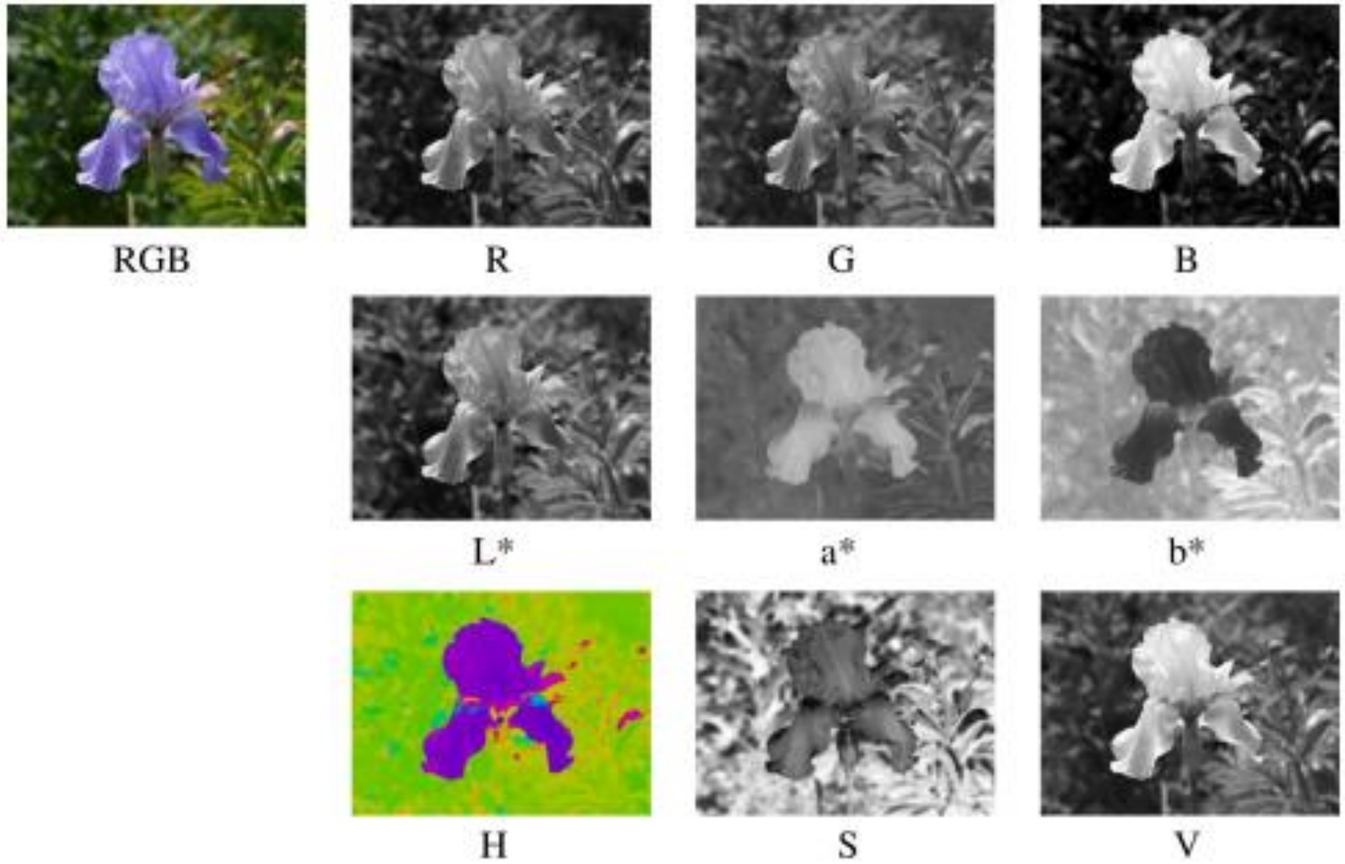


- Each pixel integrates the light spectrum L according to its spectral sensitivity S:

$$R = \int L(\lambda)\, S_R(\lambda)\, d\lambda$$

- The spectral response curves are provided by the camera manufacturer

Color Spaces



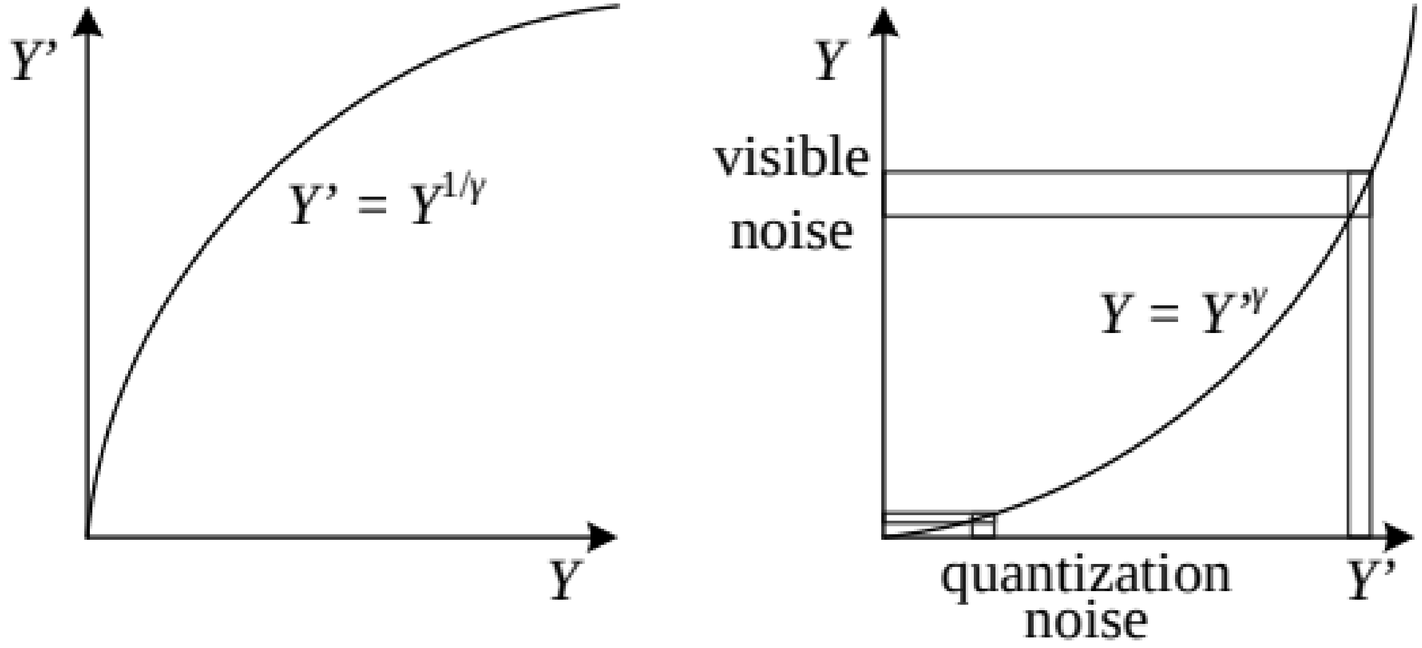
- Various different color spaces have been developed and are used in practice

Color Spaces

Various different color spaces have been developed and are used in practice :

- Physical approach: RGB, XYZ,...(linear spaces).
- Visual approach: Munsell, HSV, HSL ...(non-linear spaces).
- Physical approach with psychometry: Lab, Luv,...(non-linear spaces).

Gamma Compression

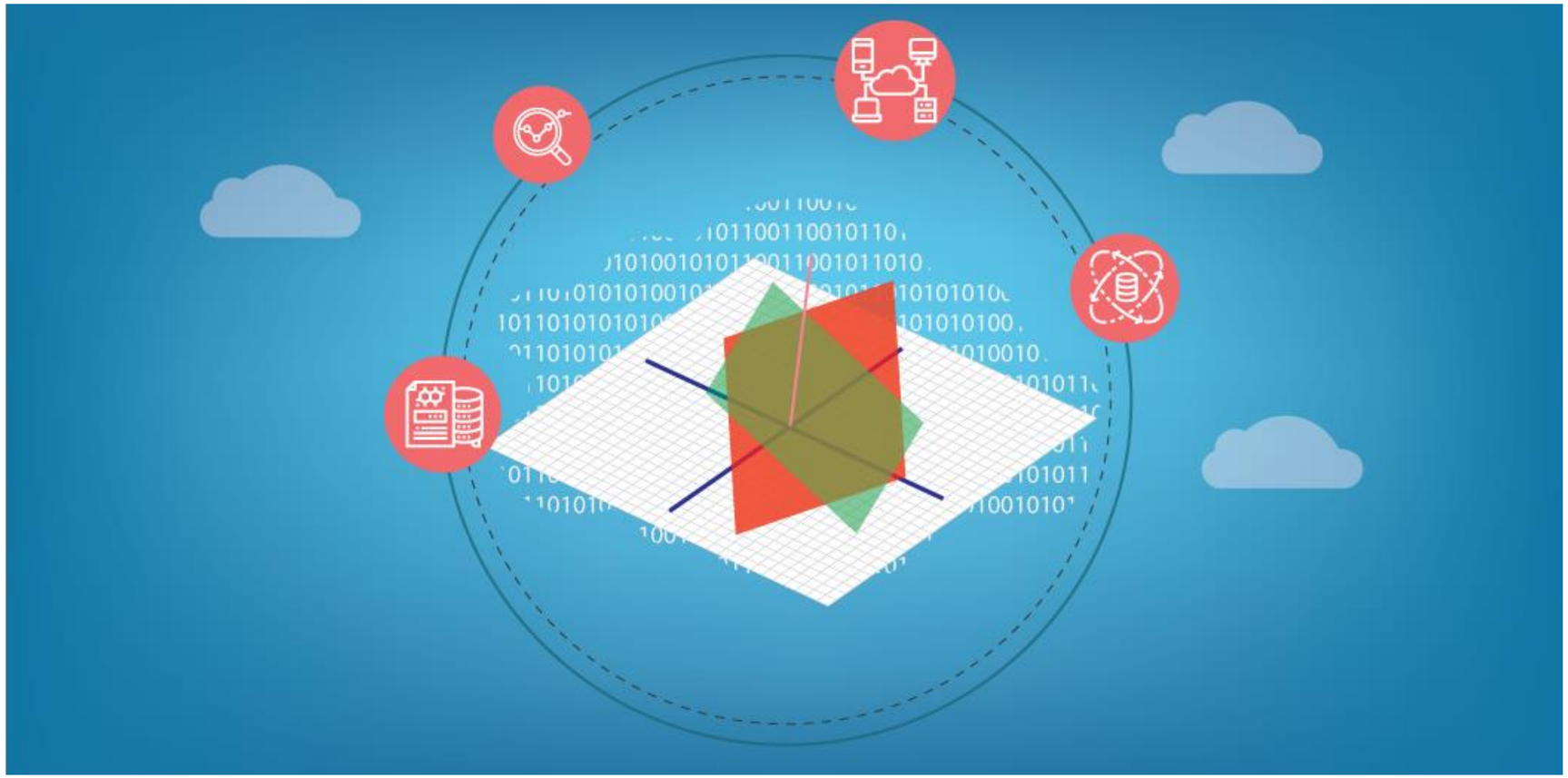


- Humans are more sensitive to intensity differences in darker regions
- Therefore, it is beneficial to nonlinearly transform (left) the intensities or colors prior to discretization (left) and to undo this transformation during loading

Image Compression

- Typically luminance is compressed with higher fidelity than chrominance
- Often, (8X8 pixel) patch-based discrete cosine or wavelet transforms are used
- Discrete Cosine Transform (DCT) is an approximation to PCA on natural images
- The coefficients are quantized to integers that can be stored with Huffman codes
- More recently, deep network based compression algorithms are developed.

Linear Algebra Prerequisites



Some Prerequisites

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous Coordinates
  - Translation
  - Scaling
- Camera Model

Some Prerequisites

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Transformation Matrices
  - Homogeneous Coordinates
  - Translation
  - Scaling
- Camera Model

Vectors and matrices are just collections of numbers that represent something: movements in space, scaling factors, pixel brightnesses, etc. We will define some common uses and standard operations on them.

Vector

- A column vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$

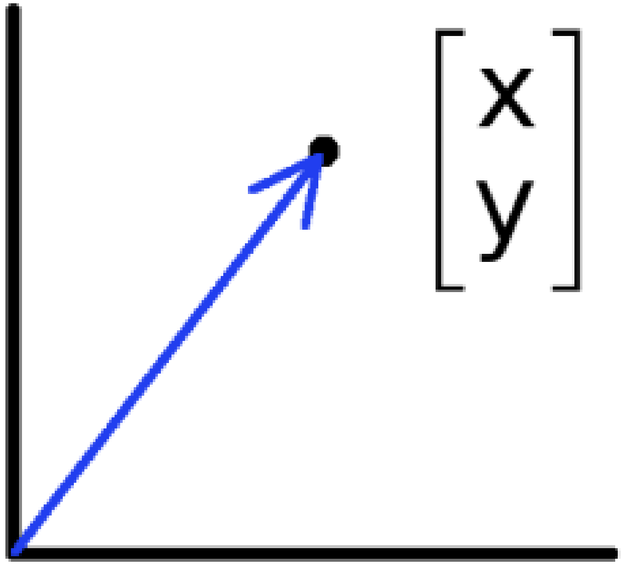
where $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

- A row vector $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$

where $\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$

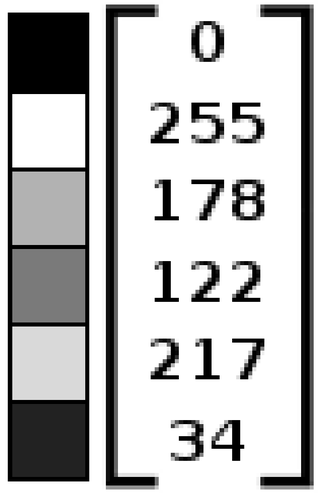
T denotes the transpose operation

Vector



Data (pixels, gradients at an image keypoint, …) can also be treated as a vector

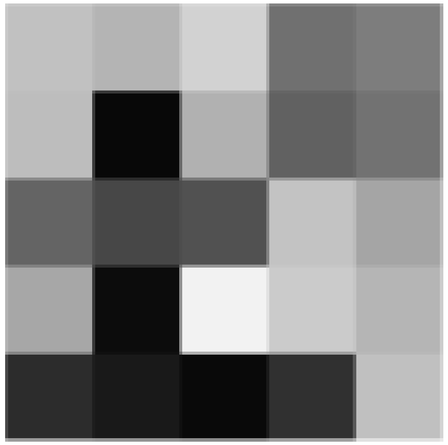
Such vectors don't have a geometric interpretation, but calculations like "distance" can still have value.



Vectors can represent an offset in 2D or 3D space.

Points are just vectors from the origin

Images



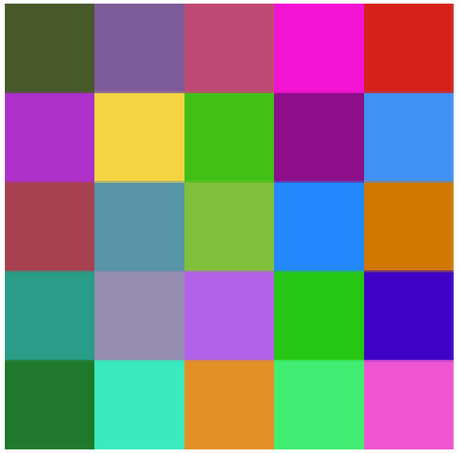
$$=\begin{bmatrix} 193 & 180 & 210 & 112 & 125 \\ 189 & 8 & 177 & 97 & 114 \\ 100 & 71 & 81 & 195 & 165 \\ 167 & 12 & 242 & 203 & 181 \\ 44 & 25 & 9 & 48 & 192 \end{bmatrix}$$

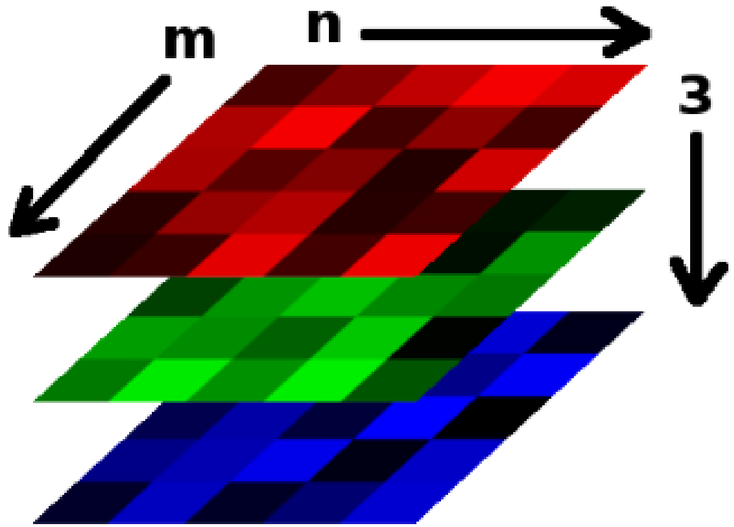
Images are represented by matrices of pixel brightness.

The higher the value of the matric is the brighter the image is.

Color Images

- Grayscale image have one number per pixel, and are stored as an mxn matrix.
- Color images have 3 numbers per pixel - red, green and blue brightness, stored as an mxnx3.
- …..





Images

Images are represented by matrices of pixel brightness.

The I (i ; j) value of a pixel s = (i ; j) represents its intensity (or color)

In **grayscale :**

- binary: I (i ; j) = 0 black or I (i ; j) = 1 white
- 8-bit coding (most common): I (i ; j) = 0; :::; 255 from darkest to lightest.

In color :

- coding in RGB space: three light intensities red, green, blue.
- 24-bit coding (the most classic): R(i ; j) = 0; :::; 255 ;G (i ; j) = 0; :::; 255 ; B(i ; j) = 0; :::; 255

Matrix Operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

- We can add a matrix with matching dimension, or a scalar.

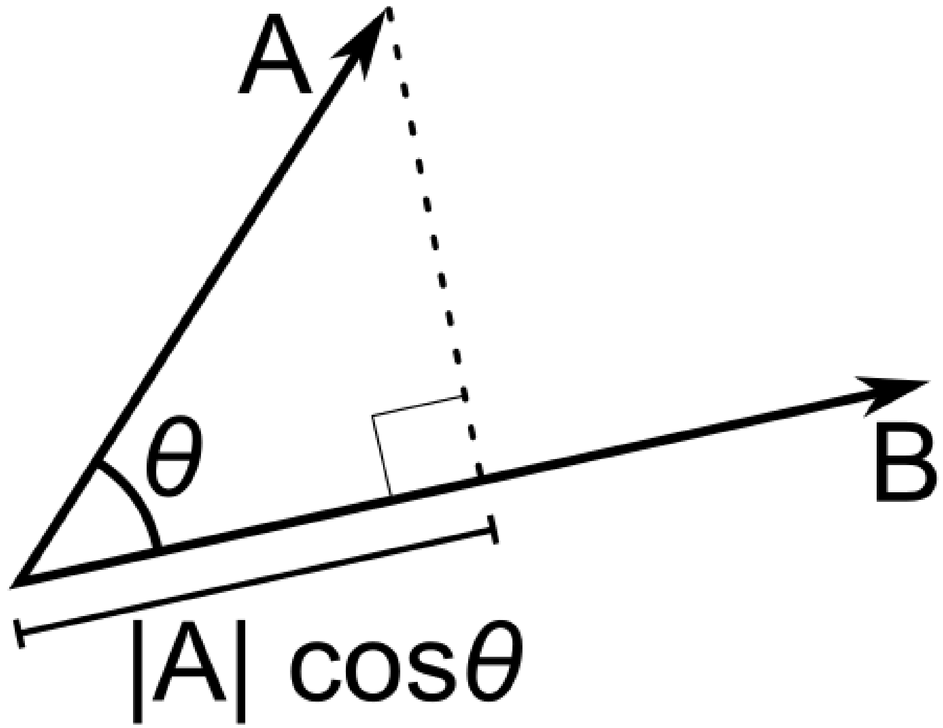
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

Matrix Operations

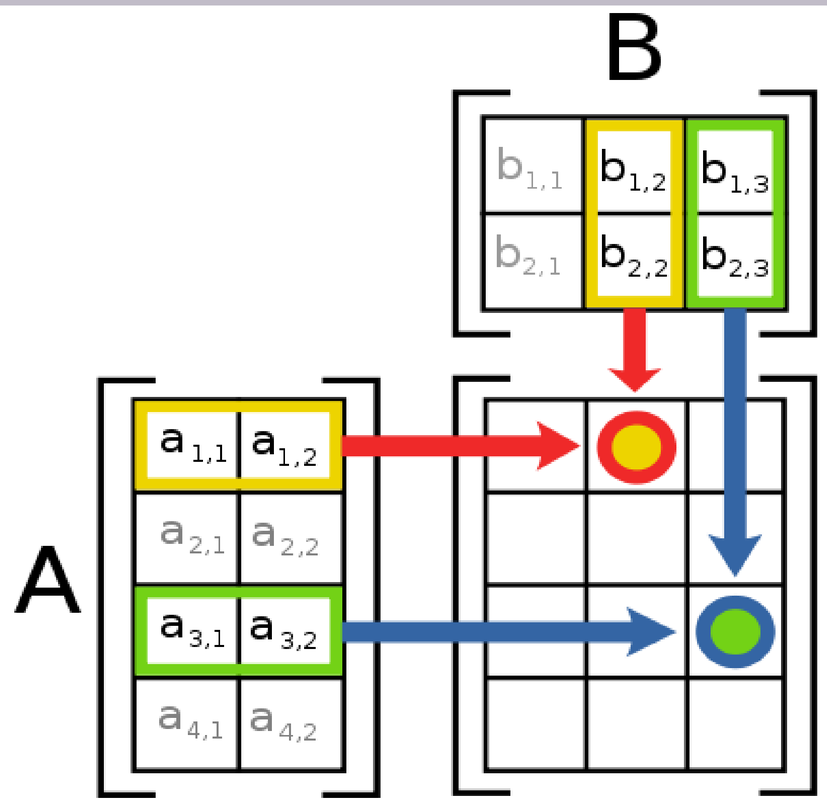
- Inner product (dot product) of vectors
  - If B is a unit vector, then AB gives the length of A which lies in the direction of B.



Matrix Operations

- Multiplication

- The product AB is:



Each entry in the result is (that row of A) dot product with (that column of B)

Multiplication has a lot of uses.

Special Matrices

- Identity matrix I
  - Square matrix, 1's along diagonal, 0's elsewhere
  - I [another matrix] = [that matrix]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Diagonal matrix
  - Square matrix with numbers along diagonal, 0's elsewhere
  - A diagonal [another matrix] scales the rows of that matrix

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

- Symmetric matrix

$$\mathbf{A}^T = \mathbf{A} \qquad \begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 7 \\ 5 & 7 & 1 \end{bmatrix}$$

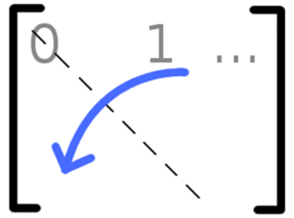
- Skew-symmetric matrix

$$\mathbf{A}^T = -\mathbf{A} \qquad \begin{bmatrix} 1 & -2 & -5 \\ 2 & 1 & -7 \\ 5 & 7 & 1 \end{bmatrix}$$

Special Matrices

- Transpose - flip matrix, so row 1 becomes column 1

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$



- A useful identity:

$$(ABC)^T = C^T B^T A^T$$

- Inverse matrix: $AA^{-1} = I$
- A useful property: $(AB)^{-1} = B^{-1}A^{-1}$

Transformations

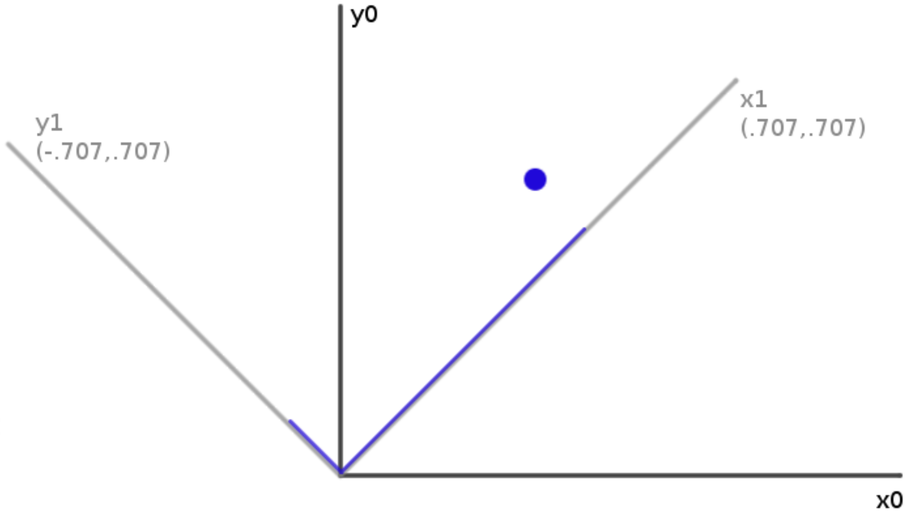
Transformations

- Matrices can be used to transform vectors in useful ways, through multiplication.
- Simplest is scaling:

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$$

Transformations

- Rotation
  - How can we convert a vector represented in frame 0 to a new rotated frame 1?
  - We must produce this vector: [component in direction of new x axis, component in direction of new y axis]
  - We can do this easily with dot products!
    - New x coordinate is [original vector] dot [the new x axis]
    - New y coordinate is [original vector] dot [the new y axis]



R x p =
rotated p'

$$p = \begin{bmatrix} px \\ py \end{bmatrix}$$

$$R = \begin{bmatrix} .707 & .707 \\ -.707 & .707 \end{bmatrix}$$

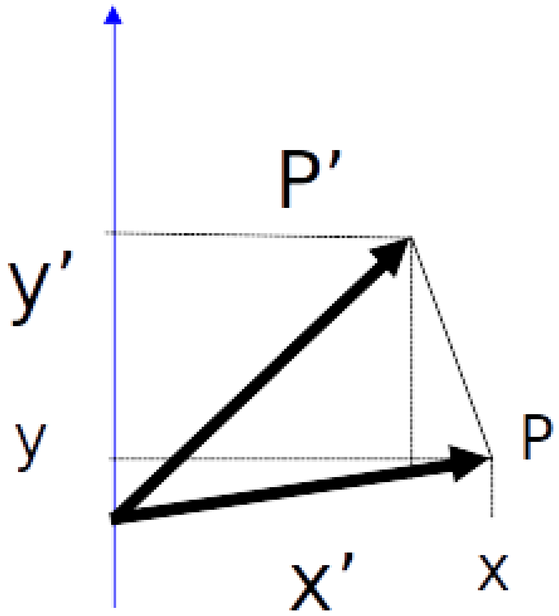
$$\begin{bmatrix} px' \\ py' \end{bmatrix}$$

Transformations

- 2D Rotation Matrix Formula
  - Counter-clockwise rotation by an angle
  - Orthonormal/ orthogonal matrices $R^{-1} = R^T$

$$x' = \cos\theta\ x - \sin\theta\ y$$

$$y' = \cos\theta\ y + \sin\theta\ x$$



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P'} = \mathbf{R}\ \mathbf{P}$$

Transformations

- Multiple transformation matrices can be used to transform a point:

  p’ = R2 R1 S p

- The effect of this is to apply their transformations one after the other, from right to left.
- In the example above, the results is (R2(R1(S p)))
- The result is exactly the same if we multiply the matrices first, to form a single transformation matrix: p’ =(R2 R1 S) p

Transformations

- In general a matrix multiplication lets us linearly combine components of a vector.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

- This is sufficient for scale, rotate, skew transformations.
- What about translation?

Transformations

- Add a ‘1’ at the end of every vector:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

- Now, we can rotate, scale and skew like before, AND translate
- This is called “homogeneous coordinates”.

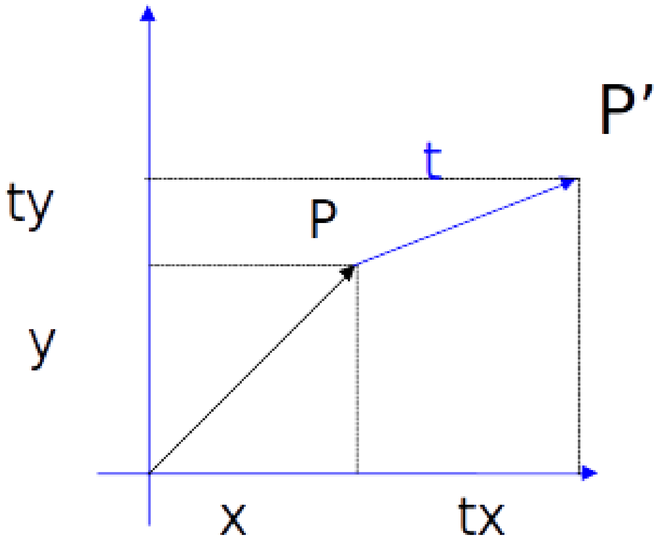
Transformations

- Homogeneous system
  - Let's assume we want to divide the result by something.
    - For example, we may want to divide by a coordinate, to make things scale down as they get further away in a camera image.
    - So **by convention**,in homogeneous coordinates, we will divide the result by its last coordinate after doing a matrix multiplication.

$$\begin{bmatrix} x \\ y \\ 7 \end{bmatrix} \Rightarrow \begin{bmatrix} x/7 \\ y/7 \\ 1 \end{bmatrix}$$

Transformations

- 2D Translation using Homogeneous Coordinates



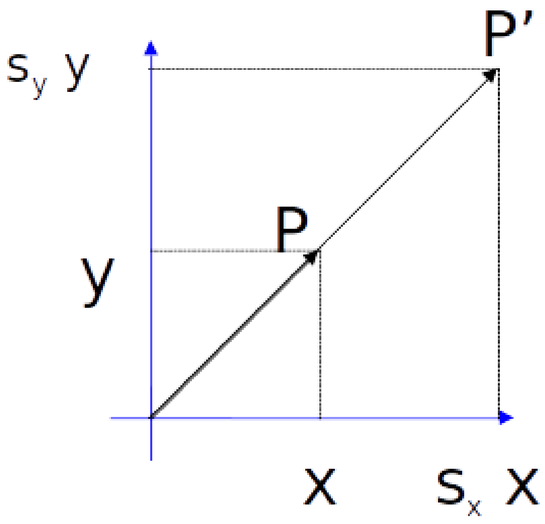
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformations

- 2D Scaling using Homogeneous Coordinates



$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}' & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \mathbf{P} = \mathbf{S} \cdot \mathbf{P}$$

Transformations

- 2D Scaling & Translation using Homogeneous Coordinates

$$\mathbf{P}'' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformations

- Translation & Scaling != Scaling & Translation

$$\mathbf{P'''} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

$$\mathbf{P'''} = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

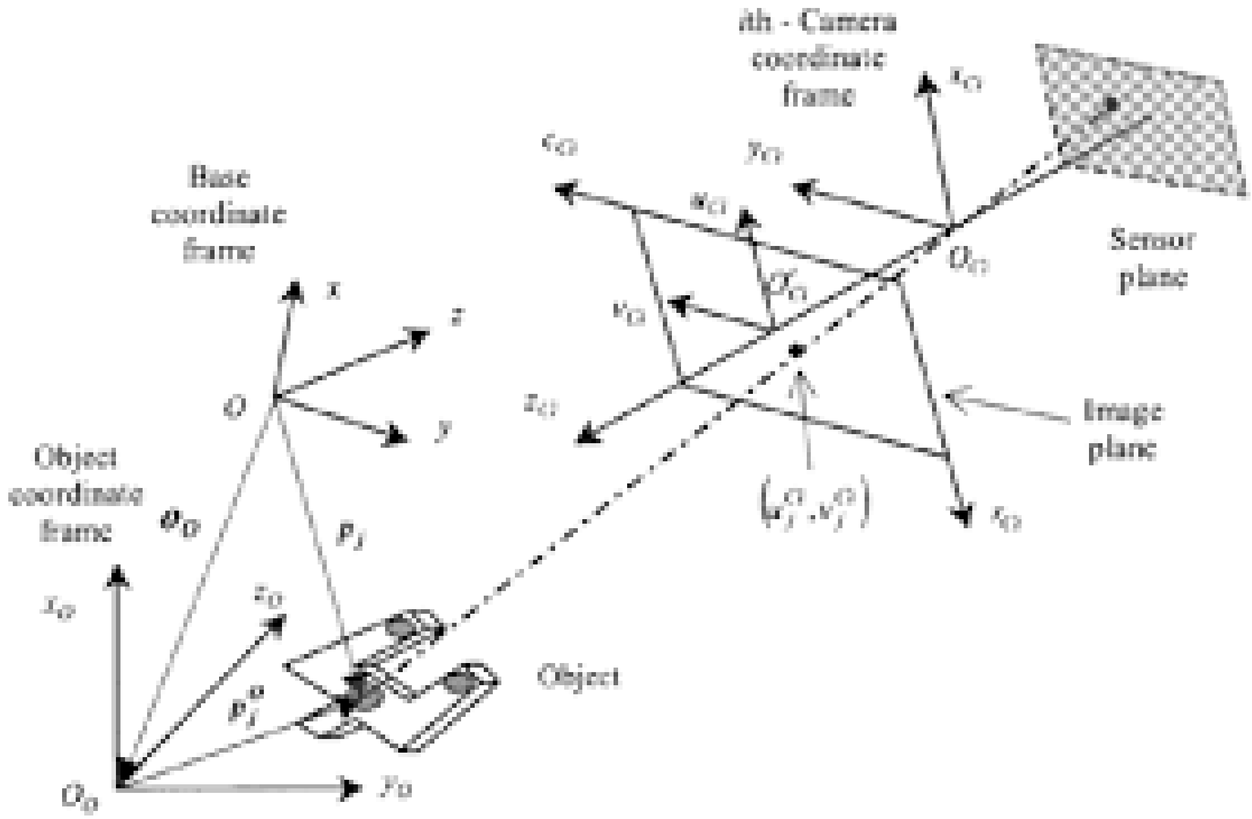
Transformations

- Rotation & Translation & Scaling

$$P' = (T\ R\ S)\ P$$

$$\mathbf{P'} = \mathbf{T} \cdot R \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

System of the Camera



Recap

- Computer Vision is a huge field
  - It can impact every aspect of life and society
  - It will drive the next information and AI revolution
  - Images is all around our lives
- Computer Vision is a highly technical field
  - Needs coding
  - Needs math understanding
  - Needs critical thinking

Next Lecture

- Lecture: image filtering

- Lab: implementation of basic filtering in Python
  - The notebook and the exercises will be posted on edunao
  - Bring your laptop