ME336 Collaborative Robot Learning
Spring 2023

Lecture 03
Perception and Control

Song Chaoyang
Southern University of Science and Technology

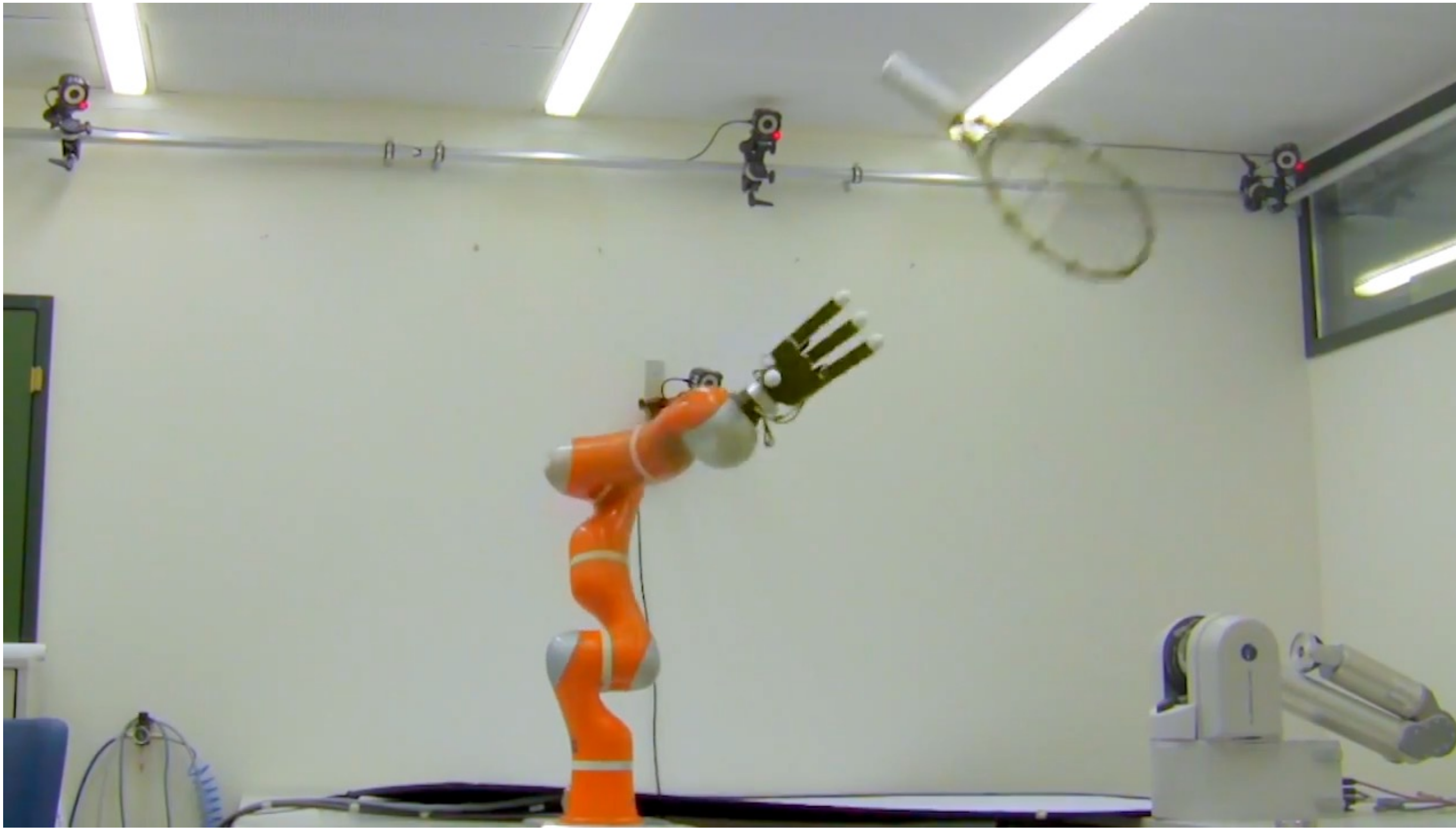

ROBOT JOINTS
Celera

What is
Robot Perception?

Making sense
of the unstructured, real, physical world

A Dynamical System Approach for Softly Catching a Flying Object: Theory and Experiment

10.1109/TRO.2016.2536749



Perception is the organization, identification, and interpretation of **sensory information** in order to represent and understand the presented **information or environment**.

Structural digitization
of the unstructured, real, physical world

Information Theory is the scientific study of the quantification, storage, and communication of **digital information**.

Digital information can be

- rapidly duplicated and easily distributed
- stored in multiple locations
- created and communicated automatically
- stored with varying levels of "discoverability"

Physical information has a fixed position in place and time.

Physical Human-Robot Collaboration

3 nested layers of consistent behaviors that the robot must follow to achieve safe pHRI

doi: 10.3389/fnbot.2020.576846

- **Safety**
  - *The first and most important feature in collaborative robots*
  - Generally addressed through **collision avoidance** (with both humans or obstacles), a feature that requires ***high reactivity (high bandwidth)*** and ***robustness*** at both perception and control layers.

Physical Human-Robot Collaboration

3 nested layers of consistent behaviors that the robot must follow to achieve safe pHRI

doi: 10.3389/fnbot.2020.576846

- **Safety**
  - *The first and most important feature in collaborative robots*
  - Generally addressed through **collision avoidance** (with both humans or obstacles), a feature that requires ***high reactivity (high bandwidth)*** and ***robustness*** at both perception and control layers.
- **Coexistence**
  - *The robot capability of sharing the workspace with humans*
  - This includes applications involving **a passive human** (e.g., medical operations where the robot is intervening on the patients' body), as well as scenarios where **robot and human work together on the same task**, without contact or coordination.

Physical Human-Robot Collaboration

3 nested layers of consistent behaviors that the robot must follow to achieve safe pHRI

doi: 10.3389/fnbot.2020.576846

- **Safety**
  - *The first and most important feature in collaborative robots*
  - Generally addressed through **collision avoidance** (with both humans or obstacles), a feature that requires ***high reactivity (high bandwidth)*** and ***robustness*** at both perception and control layers.
- **Coexistence**
  - *The robot capability of sharing the workspace with humans*
  - This includes applications involving **a passive human** (e.g., medical operations where the robot is intervening on the patients' body), as well as scenarios where **robot and human work together on the same task**, without contact or coordination.
- **Collaboration**
  - *The capability of performing robot tasks with direct human interaction and coordination*
  - **Physical collaboration** (with explicit and intentional contact between human and robot), and
  - **Contactless collaboration** (where the actions are guided by an exchange of information, e.g., in the form of body gestures, voice commands, or other modalities).
    - Establish means for ***intuitive control*** by the human operators, which may be ***non-expert users***.
    - The robot should be ***proactive*** in realizing the requested tasks, and it should be capable of ***inferring the user's intentions***, to ***interact more naturally*** from the human viewpoint.

Physical Human-Robot Collaboration

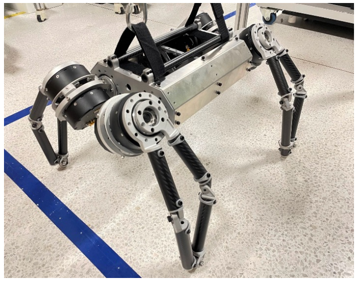
3 nested layers of consistent behaviors that the robot must follow to achieve safe pHRI

doi: 10.3389/fnbot.2020.576846

- **Safety**
  - *The first and most important feature in collaborative robots*
  - Generally addressed through **collision avoidance** (with both humans or obstacles), a feature that requires ***high reactivity (high bandwidth)*** and ***robustness*** at both perception and control layers.
- **Coexistence**
  - *The robot capability of sharing the workspace with humans*
  - Th… the robot is int… **together on the same task**, without contact or coordination.
- **Collaboration**
  - *The capability of performing robot tasks with direct human interaction and coordination*
  - **Physical collaboration** (with explicit and intentional contact between human and robot), and
  - **Contactless collaboration** (where the actions are guided by an exchange of information, e.g., in the form of body gestures, voice commands, or other modalities).
    - Establish means for ***intuitive control*** by the human operators, which may be ***non-expert users***.
    - The robot should be ***proactive*** in realizing the requested tasks, and it should be capable of ***inferring the user's intentions***, to ***interact more naturally*** from the human viewpoint.

The unpredictability of human actions

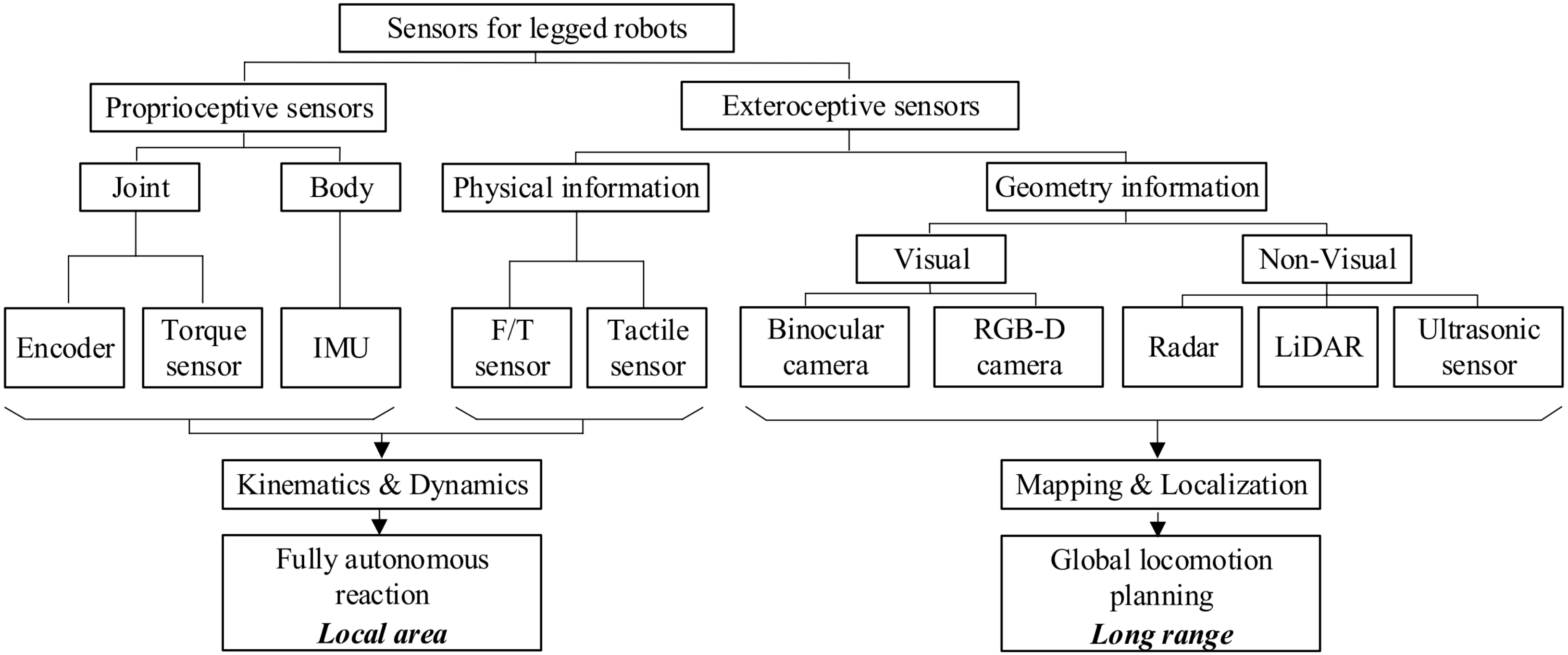
Common Sensors in Robots



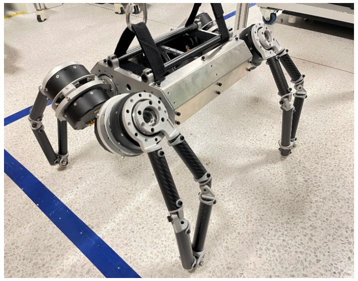
https://doi.org/10.1186/s10033-020-00485-9

Proprioceptive vs. Exteroceptive

Legged robots as an example



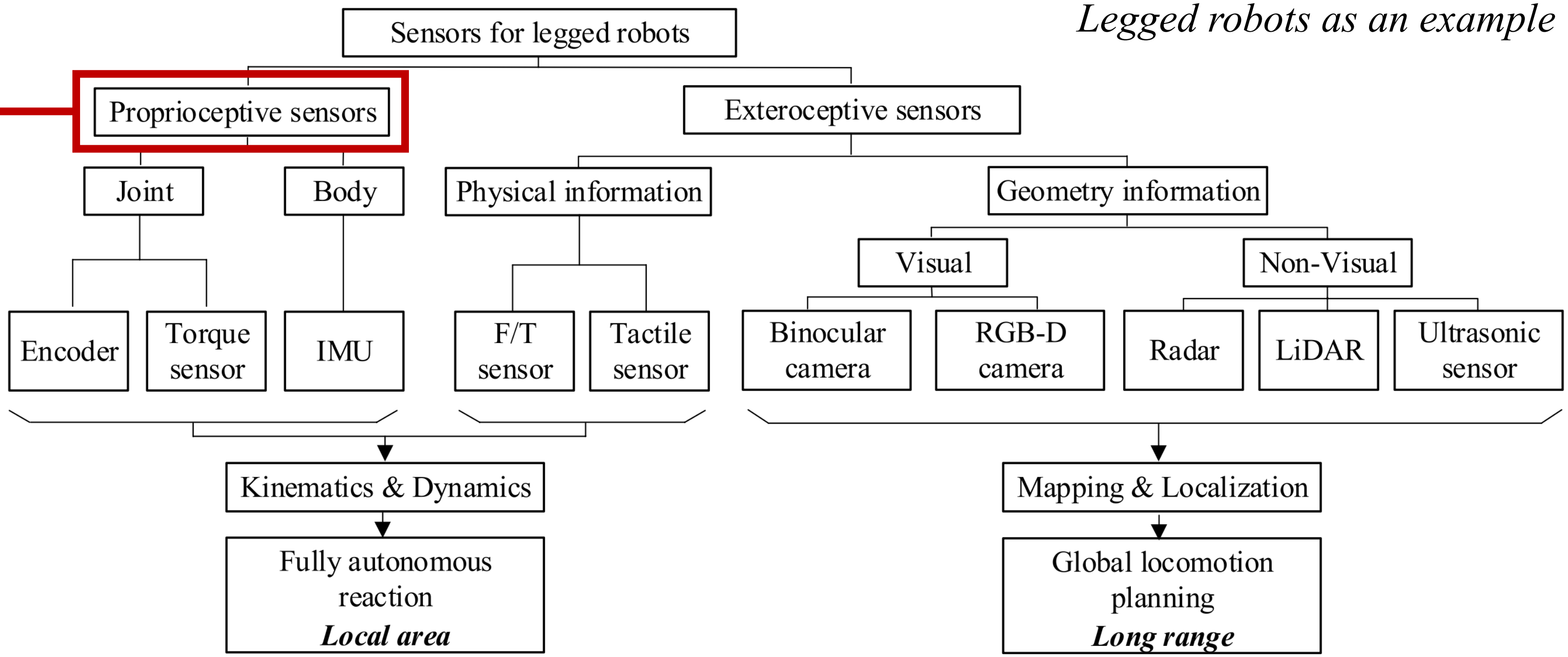
Common Sensors in Robots



Proprioceptive sensors

https://doi.org/10.1186/s10033-020-00485-9

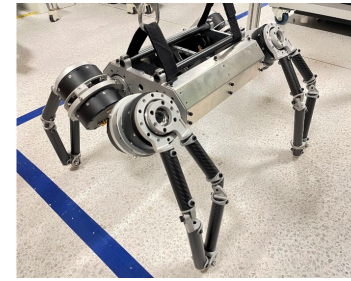
Legged robots as an example



Sense states inside the robot (e.g., joint angle, speed, torque)

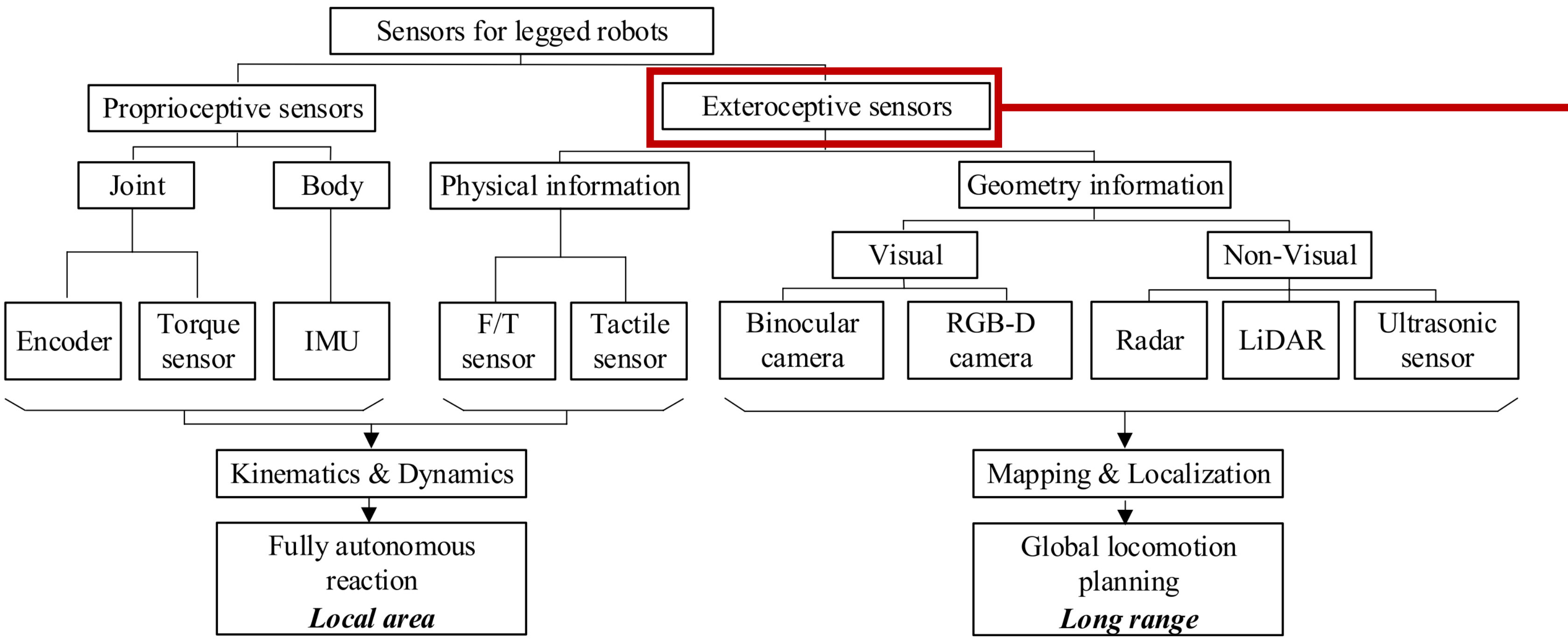
- Used by the robot control systems to receive feedback on the execution of motion and in general on the status of the robot.

Common Sensors in Robots



Exteroceptive sensors

https://doi.org/10.1186/s10033-020-00485-9



Sense states outside the robot (e.g., proximity, vision)

- Provide the robot control system information
- About the environment around the robot (e.g., rover Cameras provide images of the terrain around the rover) and
- About the effect of robot actions on the environment (e.g., the distance between a robot hand and the object it grasps)

Sensing Modalities for Control

Distance

doi: 10.3389/fnbot.2020.576846

Sensing Modalities Explained

Further Details for Reading

doi: 10.3389/fnbot.2020.576846

- **Vision**. This includes methods for processing and understanding images, to produce numeric or symbolic information reproducing human sight.
  - Although image processing is complex and computationally expensive, the richness of this sense is unique. Robotic vision is fundamental for understanding the environment and human intention, so as to react accordingly.
- **Touch**. Here, touch includes both proprioceptive force and tact, with the latter involving direct physical contact with an external object.
  - Proprioceptive force is analogous to the sense of muscle force. The robot can measure it either from the joint position errors or via torque sensors embedded in the joints; it can then use both methods to infer and adapt to human intentions, by relying on force control.
  - Human tact, on the other hand, results from activation of neural receptors, mostly in the skin. These have inspired the design of artificial tactile skins, thoroughly used for human-robot collaboration.
- **Audition**. In humans, localization of sound is performed by using binaural audition (i.e., two ears).
  - By exploiting auditory cues in the form of level/time/phase differences between left and right ears we can determine the source's horizontal position and its elevation. Microphones artificially emulate this sense, and allow robots to "blindly" locate sound sources. Although robotic hearing typically uses two microphones mounted on a motorized head, other non- biological configurations exist, e.g., a head instrumented with a single microphone or an array of several omni-directional microphones.
- **Distance**. This is the only sense among the four that humans cannot directly measure.
  - Yet, numerous examples exist in the mammal kingdom (e.g., bats and whales), in the form of echolocation. Robots measure distance with optical (e.g., infrared or lidar), ultrasonic, or capacitive (Göger et al., 2010) sensors. The relevance of this particular "sense" in human-robot collaboration is motivated by the direct relationship existing between the distance from obstacles (here, the human) and safety.

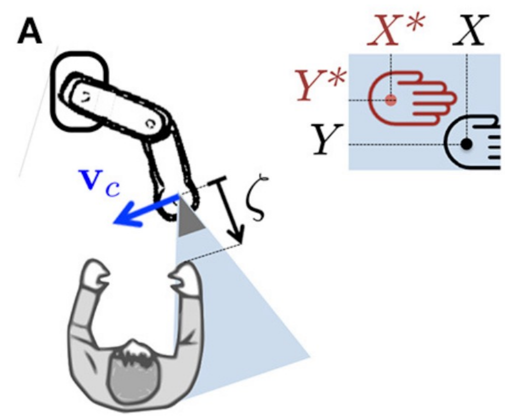
Sensor-Based Control

Basic Formulation

doi: 10.3389/fnbot.2020.576846

- Sensor-based control aims at deriving the robot control input $\boldsymbol{u}$ that minimizes a trajectory error $\boldsymbol{e} = \boldsymbol{e}(\boldsymbol{u})$, which can be estimated by sensors and depends on $\boldsymbol{u}$.
  - $\boldsymbol{u}$: operational space velocity, joint velocity, displacement, etc.

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2} \|\boldsymbol{e}(\boldsymbol{u})\|^2$$



- A general way of formulating this controller is as the quadratic minimization problem
  - actuation redundancy $\dim(\boldsymbol{u}) > \dim(\boldsymbol{e})$,
  - sensing redundancy $\dim(\boldsymbol{u}) < \dim(\boldsymbol{e})$, and
  - task constraints

Sensor-Based Control

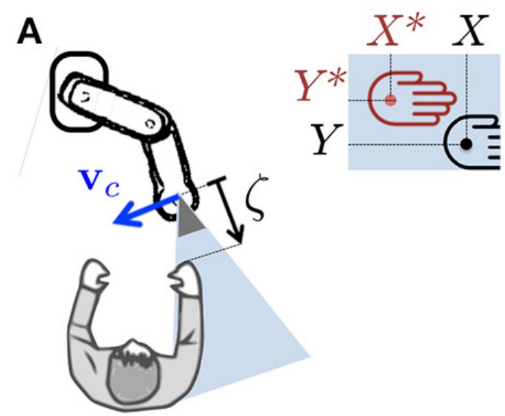
Basic Formulation

doi: 10.3389/fnbot.2020.576846

- Sensor-based control aims at deriving the robot control input $\boldsymbol{u}$ that minimizes a trajectory error $\boldsymbol{e} = \boldsymbol{e}(\boldsymbol{u})$, which can be estimated by sensors and depends on $\boldsymbol{u}$.
  - $\boldsymbol{u}$: operational space velocity, joint velocity, displacement, etc.

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2} \|\boldsymbol{e}(\boldsymbol{u})\|^2$$

- Inverse Kinematics Problem
  - Let $\boldsymbol{u} = \dot{\boldsymbol{q}}$, Given $\boldsymbol{x}$,
  - Solve: a designed value of $\boldsymbol{x}^*$



Controlling the robot joint velocities $\dot{\boldsymbol{q}}$, so that the end-effector operational space position $\boldsymbol{x}$ converges to a desired value $\boldsymbol{x}^*$

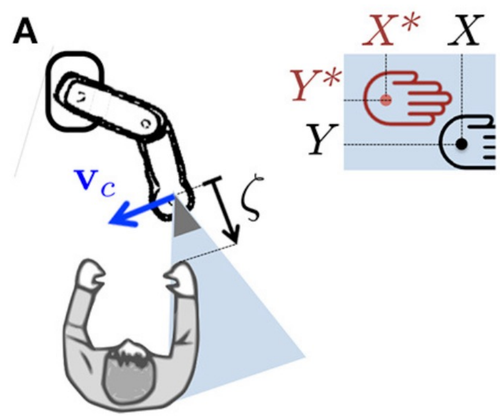
Sensor-Based Control

Basic Formulation

doi: 10.3389/fnbot.2020.576846

- Sensor-based control aims at deriving the robot control input $\boldsymbol{u}$ that minimizes a trajectory error $\boldsymbol{e} = \boldsymbol{e}(\boldsymbol{u})$, which can be estimated by sensors and depends on $\boldsymbol{u}$.
    - $\boldsymbol{u}$: operational space velocity, joint velocity, displacement, etc.

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2} \|\boldsymbol{e}(\boldsymbol{u})\|^2$$



- Inverse Kinematics Problem
    - Let $\boldsymbol{u} = \dot{\boldsymbol{q}}$, Given $\boldsymbol{x}$,
    - Solve: a designed value of $\boldsymbol{x}^*$
    - Define: $\dot{\boldsymbol{x}}^* = -\lambda(\boldsymbol{x} - \boldsymbol{x}^*)$ as desired end-effector rate ($\lambda > 0$)
    - Set: $\boldsymbol{e} = \boldsymbol{J}\dot{\boldsymbol{q}} - \dot{\boldsymbol{x}}^*$ ($\boldsymbol{J} = \partial \boldsymbol{x}/\partial \boldsymbol{q}$: Jacobian matrix)

λ is a positive tuning scalar that determines the convergence rate of task error $\boldsymbol{e}$ to 0

Sensor-Based Control

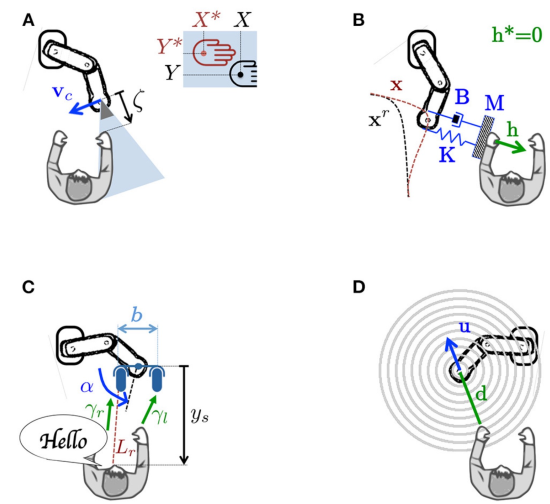
Basic Formulation

doi: 10.3389/fnbot.2020.576846

- Sensor-based control aims at deriving the robot control input $\boldsymbol{u}$ that minimizes a trajectory error $\boldsymbol{e} = \boldsymbol{e}(\boldsymbol{u})$, which can be estimated by sensors and depends on $\boldsymbol{u}$.
    - $\boldsymbol{u}$: operational space velocity, joint velocity, displacement, etc.

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2}\|\boldsymbol{e}(\boldsymbol{u})\|^2$$

- Inverse Kinematics Problem
    - Let $\boldsymbol{u} = \dot{\boldsymbol{q}}$, Given $\boldsymbol{x}$,
    - Solve: a designed value of $\boldsymbol{x}^*$
    - Define: $\dot{\boldsymbol{x}}^* = -\lambda(\boldsymbol{x} - \boldsymbol{x}^*)$ as desired end-effector rate ($\lambda > 0$)
    - Set: $\boldsymbol{e} = \boldsymbol{J}\dot{\boldsymbol{q}} - \dot{\boldsymbol{x}}^*$ ($\boldsymbol{J} = \partial \boldsymbol{x}/\partial \boldsymbol{q}$)
- Solution: $\dot{\boldsymbol{q}} = \boldsymbol{J}^+\dot{\boldsymbol{x}}^*$
    - $\boldsymbol{J}^+$ is the generalized inverse of $\boldsymbol{J}$
    - Set-point controller: $\dot{\boldsymbol{q}} = -\boldsymbol{J}^+\lambda(\boldsymbol{x} - \boldsymbol{x}^*)$



For simplicity, we assume there are no constraints in this formulation, although off-the-shelf quadratic programming solvers could account for them.

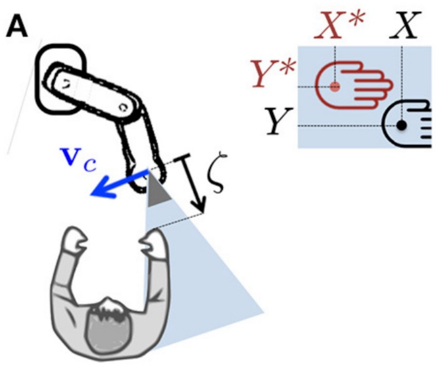
- *Nocedal, J., and Wright, S. (2000). Numerical Optimization. Springer Series in Operations Research and Financial Engineering. doi: 10.1007/b98874*

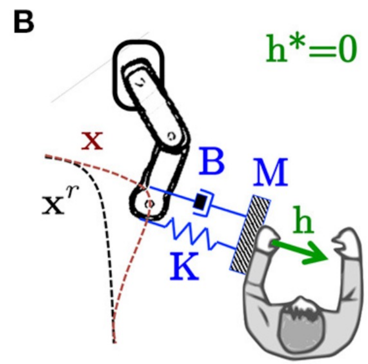
Sensor-Based Control

The four types that are commonly used in collaborative robots

doi: 10.3389/fnbot.2020.576846

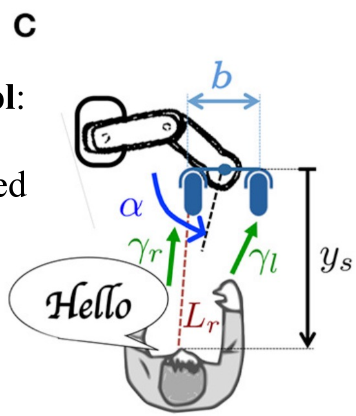
Visual servoing: the user hand is centered in the camera image.

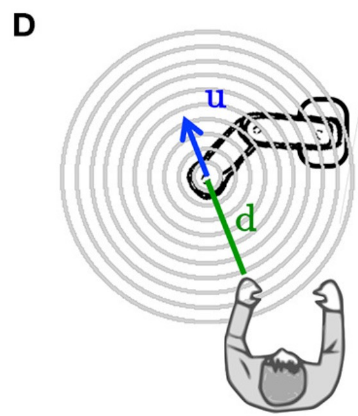




Indirect force control: by applying a wrench, the user deviates the contact point away from a reference trajectory.

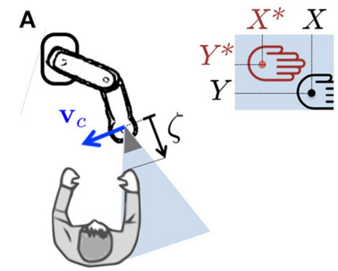
Audio-based control: a microphone rig is automatically oriented toward the sound source (the user's mouth)





Distance-based control: the user acts as a repulsive force, related to his/her distance from the robot.

Visual Servoing



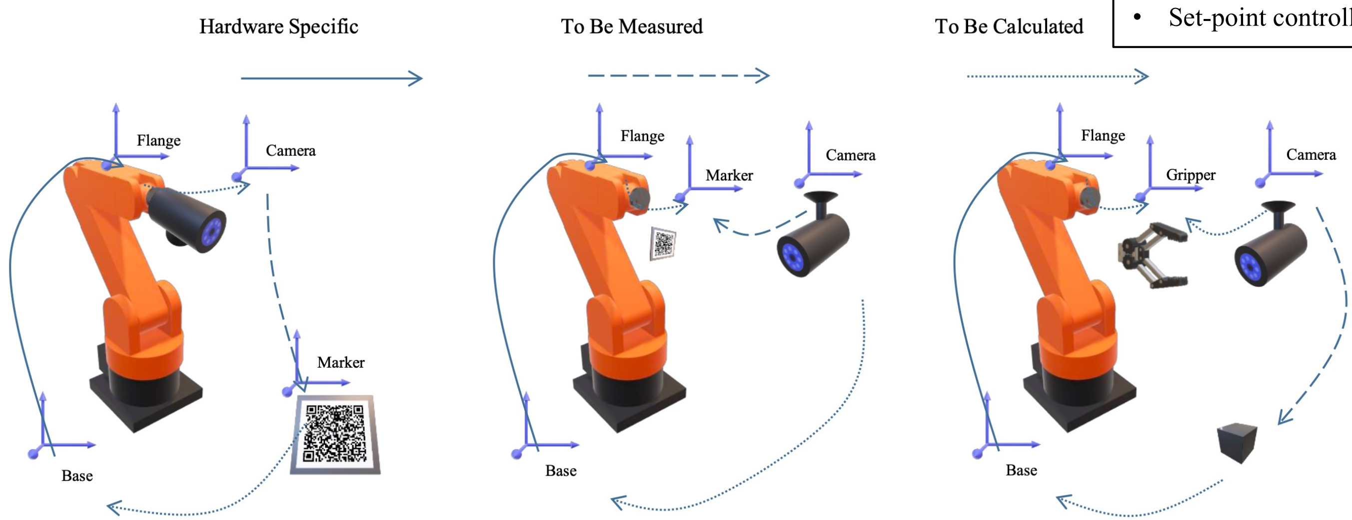
The use of vision to control robot motion

- The error $\boldsymbol{e}$ is defined with regards to some image features, here denoted by $\boldsymbol{s}$, to be regulated to a desired configuration $\boldsymbol{s}^*$
  - $\boldsymbol{s}$ is analogous to $\boldsymbol{x}$ in the inverse kinematic formulation

Inverse Kinematics Problem
- Let $\boldsymbol{u} = \dot{\boldsymbol{q}}$, Given $\boldsymbol{x}$,
- Solve: a designed value of $\boldsymbol{x}^*$
- Define: $\dot{\boldsymbol{x}}^* = -\lambda(\boldsymbol{x} - \boldsymbol{x}^*)$
- Set: $\boldsymbol{e} = \boldsymbol{J}\dot{\boldsymbol{q}} - \dot{\boldsymbol{x}}^*$ ($\boldsymbol{J} = \partial\boldsymbol{x}/\partial\boldsymbol{q}$)

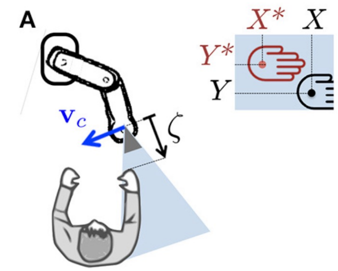
Solution: $\dot{\boldsymbol{q}} = \boldsymbol{J}^+\dot{\boldsymbol{x}}^*$
- $\boldsymbol{J}^+$ is the generalized inverse of $\boldsymbol{J}$
- Set-point controller: $\dot{\boldsymbol{q}} = -\boldsymbol{J}^+\lambda(\boldsymbol{x} - \boldsymbol{x}^*)$

- The visual error is $\boldsymbol{e} = \dot{\boldsymbol{s}} - \dot{\boldsymbol{s}}^*$



(a) Eye-in-Hand (b) Eye-on-Base (c) Learning-based

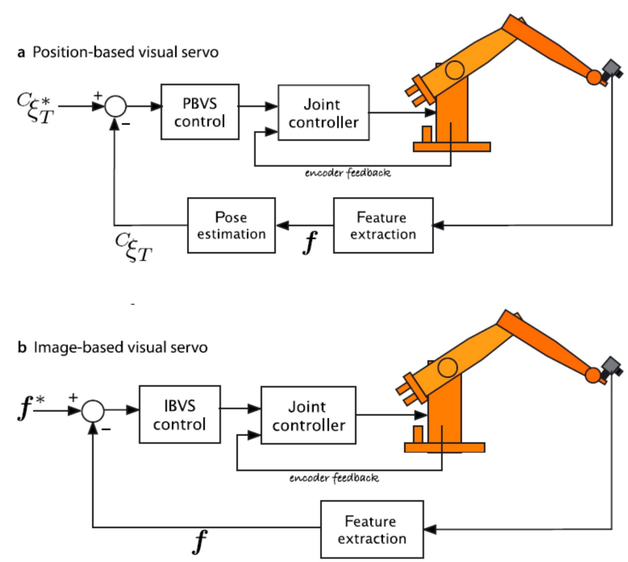
Visual Servoing



The use of vision to control robot motion

- The error $\boldsymbol{e}$ is defined with regards to some image features, here denoted by $\boldsymbol{s}$, to be regulated to a desired configuration $\boldsymbol{s}^*$
  - $\boldsymbol{s}$ is analogous to $\boldsymbol{x}$ in the inverse kinematic formulation

- The visual error is $\boldsymbol{e} = \dot{\boldsymbol{s}} - \dot{\boldsymbol{s}}^*$
  - *Position-based* if $\boldsymbol{s}$ is defined in the 3D operational space
    - Projecting the task from the image to the operational space to obtain $\boldsymbol{x}$ and then apply $\dot{\boldsymbol{q}} = -\boldsymbol{J}^+\lambda(\boldsymbol{x} - \boldsymbol{x}^*)$
  - *Image-based* if $\boldsymbol{s}$ is defined in the image space

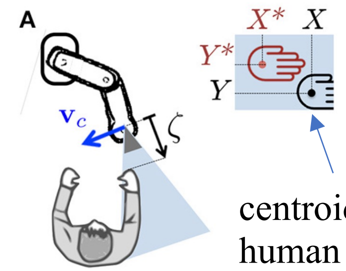


AncoraSIR.com

- Uses images, calibrated camera and known geometry model of the target to determine the pose of the target with respect to the camera.
- Control is performed in task space SE(3).

- Uses the image feature directly omitting the pose estimation step.
- Control is performed in image coordinate space R^2

Visual Servoing



The use of vision to control robot motion

- *Image-based* if $\boldsymbol{s}$ is defined in the image space
  - The visual error is $\boldsymbol{e} = \dot{\boldsymbol{s}} - \dot{\boldsymbol{s}}^*$
  - The simplest image-based controller uses $\boldsymbol{s} = [X, Y]^\top$

X and Y as the coordinates of an image pixel, to generate $\boldsymbol{u}$ that drives $\boldsymbol{s}$ to a reference $\boldsymbol{s}^* = [X^*, Y^*]^\top$

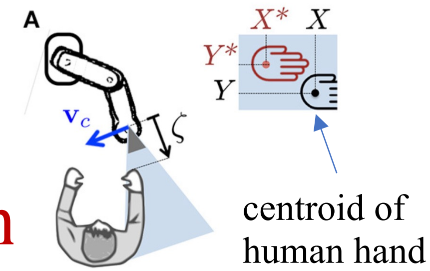
- Defining $\boldsymbol{e}$ as $\dot{\boldsymbol{s}} - \dot{\boldsymbol{s}}^* = \begin{bmatrix} \dot{X} - \dot{X}^* \\ \dot{Y} - \dot{Y}^* \end{bmatrix}$, with $\dot{\boldsymbol{s}}^* = -\lambda \begin{bmatrix} \dot{X} - \dot{X}^* \\ \dot{Y} - \dot{Y}^* \end{bmatrix}$
  - If we use the camera's 6D velocity as the control input $\boldsymbol{u} = \boldsymbol{v_c}$, the image Jacobian (*Interaction*) matrix relating $[\dot{X}, \dot{Y}]^\top$ and $\boldsymbol{u}$ is:
    - $\boldsymbol{J}_v = \begin{bmatrix} -\frac{1}{\zeta} & 0 & \frac{X}{\zeta} & XY & -1 - X^2 & Y \\ 0 & -\frac{1}{\zeta} & \frac{Y}{\zeta} & 1 + Y^2 & -XY & -X \end{bmatrix}$

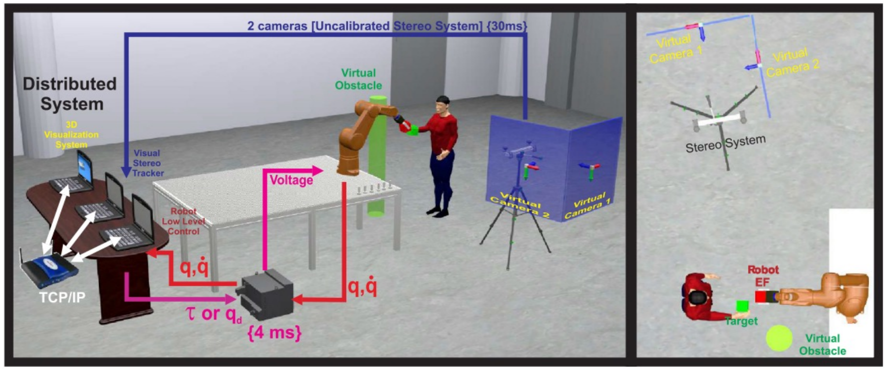
ζ denotes the depth of the point with respect to the camera

  - In the absence of constraints, the solution of $\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2} \|\boldsymbol{e}(\boldsymbol{u})\|^2$ is
    - $\boldsymbol{u} = \boldsymbol{v}_c = -\boldsymbol{J}_v^+ \lambda \begin{bmatrix} X - X^* \\ Y - Y^* \end{bmatrix}$

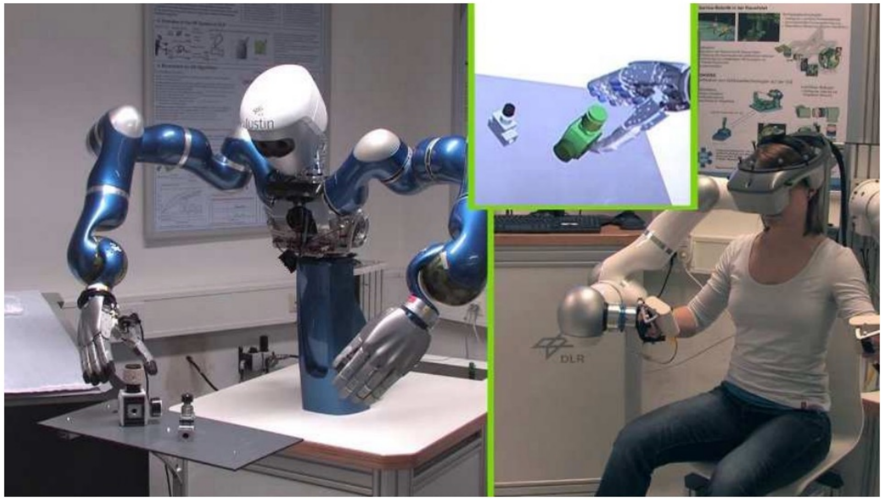
Visual Servoing

Application to Human-Robot Collaboration

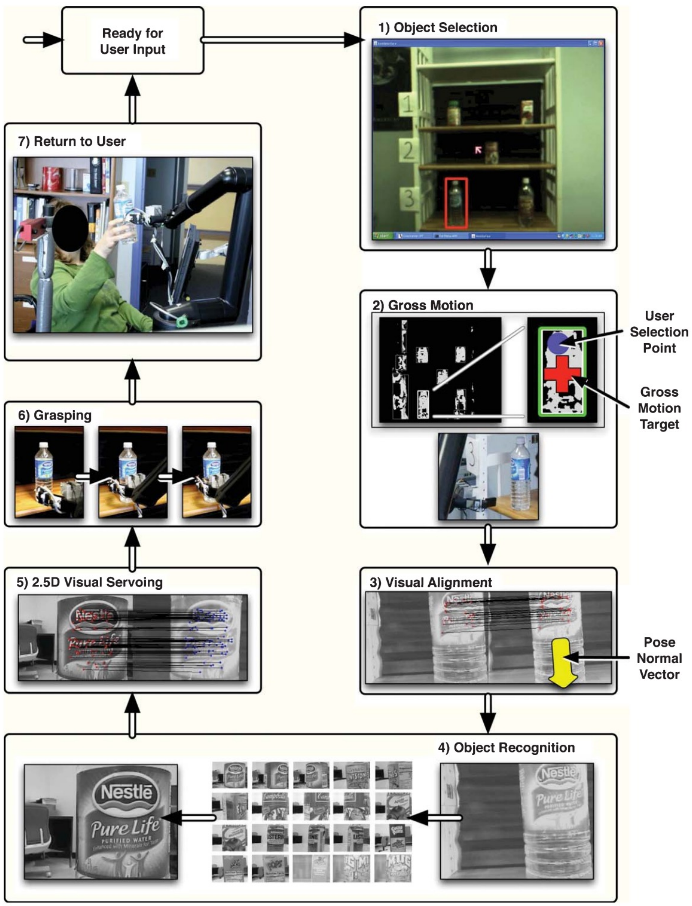




10.1109/TRO.2016.2535443

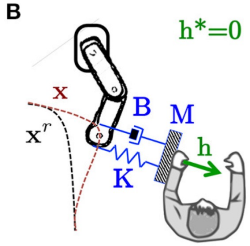


10.1109/CRV.2015.39



10.1155/2011/698079

Touch (or Force) Control



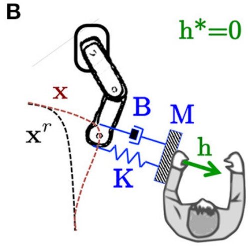
Requires measurement of one or multiple wrenches $\boldsymbol{h}$

(in the case of tactile skins)

- The measured wrenches $\boldsymbol{h}$ are (at most) composed of three translational forces, and three torques
  - $\boldsymbol{h}$ is fed to the controller that moves the robot so that it exerts a desired interaction force with the human or environment.

- Force control strategies
  - *Direct* control regulates the contact wrench to obtain a desired wrench $\boldsymbol{h}^*$.
    - Specifying $\boldsymbol{h}^*$ requires an explicit model of the task and environment.
    - i.e., Hybrid position/force control
  - *Indirect* control does not require an explicit force feedback loop.
    - Impedance control | Admittance control (Hogan, 1985)

Touch (or Force) Control



Requires measurement of one or multiple wrenches $\boldsymbol{h}$

(in the case of tactile skins)

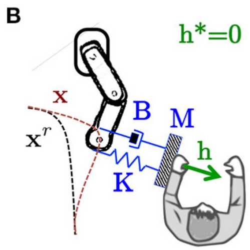
- ***Direct*** force control regulates the contact wrench to obtain a desired wrench $\boldsymbol{h}^*$.
  - Specifying $\boldsymbol{h}^*$ requires an explicit model of the task and environment.

- Hybrid position/force control, which regulates the velocity and wrench along unconstrained and constrained task directions, respectively.

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2} \|\boldsymbol{e}(\boldsymbol{u})\|^2$$

  - This is equivalent to setting $\boldsymbol{e} = \boldsymbol{S}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^*) + (\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{h} - \boldsymbol{h}^*)$
    - $\boldsymbol{S} = \boldsymbol{S}^\top \geq 0$: a binary diagonal selection matrix | $\boldsymbol{I}$: the identity matrix.

  - Applying a motion $\boldsymbol{u}$ that nullifies $\boldsymbol{e}$ guarantees that the components of $\dot{\boldsymbol{x}}$ (respectively $\boldsymbol{h}$) specified via $\boldsymbol{S}$ (respectively $\boldsymbol{I} - \boldsymbol{S}$) converge to $\dot{\boldsymbol{x}}^*$ (respectively $\boldsymbol{h}^*$).



Touch (or Force) Control

Requires measurement of one or multiple wrenches $\boldsymbol{h}$

(in the case of tactile skins)

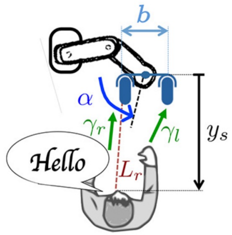
- Indirect force control does not require an explicit force feedback loop.
  - Impedance control | Admittance control (Hogan, 1985)

- Modeling the deviation of the contact point from a reference trajectory $\boldsymbol{x}^r(t)$ associated to the desired $\boldsymbol{h}^*$, via a virtual mechanical impedance with adjustable parameters
  - this is equivalent to setting $\boldsymbol{e} = \boldsymbol{M}(\ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}^r) + \boldsymbol{B}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^r) + \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}^r) - (\boldsymbol{h} - \boldsymbol{h}^*)$
    - inertia $\boldsymbol{M}$, damping $\boldsymbol{B}$, and stiffness $\boldsymbol{K}$
  - $\boldsymbol{x}$ represents the "deviated" contact point pose, with $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ as time derivatives.
    - When $\boldsymbol{e} = 0$, the displacement $\boldsymbol{x} - \boldsymbol{x}^r$ responds as a mass-spring-damping system under the action of an external force $\boldsymbol{h} - \boldsymbol{h}^*$.
    - In most cases, $\boldsymbol{x}^r(t)$ is defined for motion in free space ($\boldsymbol{h}^* = 0$).

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2}\|\boldsymbol{e}(\boldsymbol{u})\|^2$$

- The general formulation above can account for both impedance control ($\boldsymbol{x}$ is measured and $\boldsymbol{u} = \boldsymbol{h}$) and admittance control ($\boldsymbol{h}$ measured and u = x).

Audio-Based Control



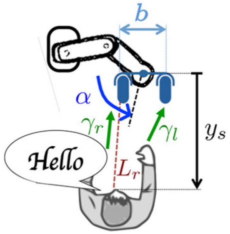
To locate the sound source and move the robot toward it.

- For simplicity, we present the two-dimensional binaural (i.e., with two microphones) configuration with the angular velocity of the microphone rig as control input: $\boldsymbol{u} = \dot{\alpha}$

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2} \|\boldsymbol{e}(\boldsymbol{u})\|^2$$

- Two popular methods for defining error $\boldsymbol{e}$
  - *Interaural **Time** Difference* (ITD) based aural servoing
    - Uses the difference τ between the arrival times of the sound on each microphone
    - τ must be regulated to a desired τ^*
  - *Interaural **Level** Difference* (ILD) based aural servoing
    - Uses ρ, the difference in intensity between the left and right signals

Audio-Based Control



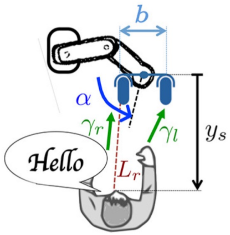
To locate the sound source and move the robot toward it.

- ***Interaural Time Difference (ITD) based aural servoing***
  - Uses the difference τ between the arrival times of the sound on each microphone; τ must be regulated to a desired τ^*

- Setting $\boldsymbol{e} = \dot{\tau} - \dot{\tau}^*$, with the desired rate $\dot{\tau}^* = -\lambda(\tau - \tau^*)$ (to obtain set-point regulation to τ^*)
  - Feature τ can be derived in real-time by using standard cross-correlation of the signals. Under a far field assumption:
    - $\boldsymbol{e} = \dot{\tau} - \dot{\tau}^* = -\left(\sqrt{(b/c)^2 - \tau^2}\right)\boldsymbol{u} - \dot{\tau}^*$
    - c the sound celerity and b the microphones baseline.

- The scalar ITD Jacobian is $\boldsymbol{J_\tau} = -\sqrt{(b/c)^2 - \tau^2}$
- The motion that minimizes $\boldsymbol{e}$ is $\boldsymbol{u} = -\lambda \boldsymbol{J}_\tau^{-1}(\tau - \tau^*)$
  - locally defined for $\alpha \in (0, \pi)$, to ensure that $|\boldsymbol{J_\tau}| \neq \boldsymbol{0}$

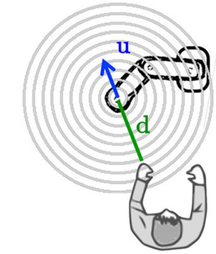
Audio-Based Control



To locate the sound source and move the robot toward it.

- ***Interaural Level Difference (ILD) based aural servoing***
  - Uses ρ, the difference in intensity between the left and right signals
- This can be obtained in a time window of size N as $\rho = \frac{E_l}{E_r}$
  - $E_{l,r} = \sum_{n=0}^{N} \gamma_{l,n}[n]^2$ denote the signals' sound energies
  - $\gamma_{l,n}[n]$ are the intensities at iteration n.
- To regulate ρ to a desired ρ^*, one can set $\boldsymbol{e} = \dot{\rho} - \dot{\rho}^*$ with $\dot{\rho}^* = -\lambda(\rho - \rho^*)$. Assuming spherical propagation and slowly varying signal:
  - $\boldsymbol{e} = \dot{\rho} - \dot{\rho}^* = \frac{y_s(\rho+1)b}{L_r^2}\boldsymbol{u} - \dot{\rho}^*$
  - y_s is the sound source frontal coordinate in the moving auditory frame
  - L_r is the distance between the right microphone and the source
- The scalar ILD Jacobian is $\boldsymbol{J}_\rho = \frac{y_s(\rho+1)b}{L_r^2}$
- The motion that minimizes $\boldsymbol{e}$ is $\boldsymbol{u} = -\lambda \boldsymbol{J}_\rho^{-1}(\rho - \rho^*)$
  - $\boldsymbol{J}_\rho^{-1}$ is defined for sources located in front of the rig.
- <u>*In contrast with ITD-servoing, here the source location (i.e., y_s and L_r) must be known or estimated.*</u>

Distance-Based Control



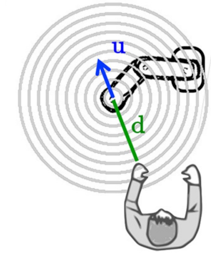
The user acts as a repulsive force, related to his/her distance from the robot

- The simplest (and most popular) distance-based controller is the artificial potential fields method
  - Despite being prone to local minima, it has been thoroughly deployed both on manipulators and on autonomous vehicles for obstacle avoidance.
  - Besides, it is acceptable that a collaborative robot stops (e.g., because of local minima) as long as it avoids the human user.

- The potential fields method consists in modeling each obstacle as a source of repulsive forces, related to the robot distance from the obstacle
  - All the forces are summed up resulting in a velocity in the most promising direction.
  - Given $\boldsymbol{d}$, the position of the nearest obstacle in the robot frame, the original version (Khatib, 1985) consists in applying operational space velocity
  - $$\boldsymbol{u} = \begin{cases} \lambda \left(\frac{1}{\|\boldsymbol{d}\|} - \frac{1}{d_0} \right) \frac{1}{\|\boldsymbol{d}\|^2} & \text{if } \|\boldsymbol{d}\| < d_0, \\ 0 & \text{otherwise} \end{cases}$$

$d_0 > 0$ is the (arbitrarily tuned) minimal distance required for activating the controller.

Distance-Based Control



The user acts as a repulsive force, related to his/her distance from the robot

- Since the quadratic denominator in $\boldsymbol{u} = \begin{cases} \lambda\left(\frac{1}{\|\boldsymbol{d}\|} - \frac{1}{d_0}\right)\frac{1}{\|\boldsymbol{d}\|^2} & \text{if } \|\boldsymbol{d}\| < d_0 \\ 0 & \text{otherwise} \end{cases}$ yields abrupt accelerations, *more recent versions adopt a linear behavior*.

- This can be obtained by setting $\boldsymbol{e} = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^*$ with $\dot{\boldsymbol{x}}^* = \lambda(1 - d_0/\|\boldsymbol{d}\|)\boldsymbol{d}$ as reference velocity
  - $\boldsymbol{e} = \dot{\boldsymbol{x}} - \lambda\left(1 - \frac{d_0}{\|\boldsymbol{d}\|}\right)\boldsymbol{d}$

- By defining as control input $\boldsymbol{u} = \dot{\boldsymbol{x}}$, the solution to $\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \frac{1}{2}\|\boldsymbol{e}(\boldsymbol{u})\|^2$ is:
  - $\boldsymbol{u} = \lambda\left(1 - \frac{d_0}{\|\boldsymbol{d}\|}\right)\boldsymbol{d}$

10.1109/ACC.1995.529274

Integration of Multiple Sensors

Integrating multiple sensors in a unique controller

- Just like natural senses, artificial senses provide complementary information about the environment.
    - Hence, to effectively perform a task, the robot should measure (and use for control) multiple feedback modalities

- Challenges to the control design,
    - e.g., sensor synchronization, task compatibility, and task representation.

- Three methods for combining N sensors within a controller
    - *Traded*: the sensors control the robot one at a time.
    - *Shared*: All sensors control the robot throughout operation.
    - *Hybrid*: the sensors act simultaneously, but on different axes of a predefined Cartesian task-frame.

Integration of Multiple Sensors

Integrating multiple sensors in a unique controller

- ***Traded***: the sensors control the robot one at a time.
  - Predefined conditions on the task trigger the switches:

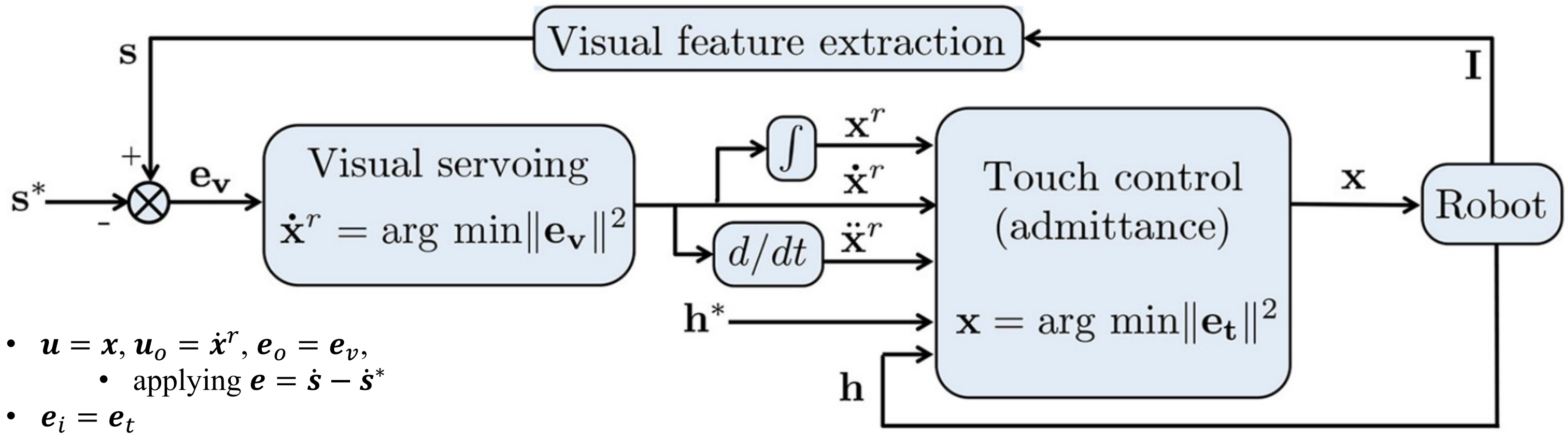
$$\boldsymbol{u} = \begin{cases} \arg\min_{\boldsymbol{u}} \|\boldsymbol{e_1}(\boldsymbol{u})\|^2 & \text{if } (condition\ 1) = \text{true} \\ \vdots & \\ \arg\min_{\boldsymbol{u}} \|\boldsymbol{e}_N(\boldsymbol{u})\|^2 & \text{if } (condition\ N) = \text{true} \end{cases}$$

10.1109/ROBOT.1998.677418
10.1109/ICRA.2014.6906917
10.1109/IROS.2013.6697019

Integration of Multiple Sensors

Integrating multiple sensors in a unique controller

- ***Shared***: All sensors control the robot throughout operation.
  - i.e., nested control loops for shared vision/touch control
  - $\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \|\boldsymbol{e_i}(\boldsymbol{u}, \boldsymbol{u_o})\|^2$ such that $\boldsymbol{u_o} = \arg\min_{\boldsymbol{u_o}} \|\boldsymbol{e_o}(\boldsymbol{u_o})\|^2$



- $\boldsymbol{u} = \boldsymbol{x}, \boldsymbol{u_o} = \dot{\boldsymbol{x}}^r, \boldsymbol{e_o} = \boldsymbol{e_v}$,
  - applying $\boldsymbol{e} = \dot{\boldsymbol{s}} - \dot{\boldsymbol{s}}^*$
- $\boldsymbol{e_i} = \boldsymbol{e_t}$
  - applying $\boldsymbol{e} = \boldsymbol{M}(\ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}^r) + \boldsymbol{B}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^r) + \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}^r) - (\boldsymbol{h} - \boldsymbol{h}^*)$

The most common scheme for shared vision/touch (admittance) control

- The goal is to obtain desired visual features $\boldsymbol{s}^*$ and wrench $\boldsymbol{h}^*$, based on current image $\boldsymbol{I}$ and wrench $\boldsymbol{h}$.
- The *outer* visual servoing loop based on error $\boldsymbol{e} = \dot{\boldsymbol{s}} - \dot{\boldsymbol{s}}^*$ outputs a reference velocity $\dot{\boldsymbol{x}}^r$ that is then deformed by the *inner* admittance control loop based on error $\boldsymbol{e} = \boldsymbol{M}(\ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}^r) + \boldsymbol{B}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^r) + \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}^r) - (\boldsymbol{h} - \boldsymbol{h}^*)$, to obtain the desired robot position $\boldsymbol{x}$.

10.1109/ROBOT.1998.677418
10.1109/ICRA.2014.6906917
10.1109/IROS.2013.6697019

Integration of Multiple Sensors

Integrating multiple sensors in a unique controller

- ***Hybrid***: the sensors act simultaneously, but on different axes of a predefined Cartesian *task-frame*.
  - The directions are selected by binary diagonal matrices $\boldsymbol{S}_j$, $j = 1, \ldots, N$ with the dimension of the task space, and such that $\sum_{j=1}^{N} \boldsymbol{S} = \boldsymbol{I}$
  - $\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \left\| \sum_{j=1}^{N} \boldsymbol{S}_j \boldsymbol{e}_j(\boldsymbol{u}) \right\|^2$
  - To express all $\boldsymbol{e}$ in the same task frame, one should apply ${}^{B}V_{A}$ and/or ${}^{B}V_{A}^{\mathrm{T}}$ when transforming 6D velocities or wrenches to a unique frame

TABLE 1 | Classification of all papers according to four criteria: sense(s) used by the robot, objective of the controller, target sector, and type of robot.

References	Sense(s)	Control objective	Sector	Robot
Cai et al. (2016) and Gridseth et al. (2016)	Vision	Contactless guidance	Service	Arm
Gridseth et al. (2015)	Vision	Remote guidance	Service	Arm
Dune et al. (2008), Tsui et al. (2011), and Narayanan et al. (2016)	Vision	Contactless guidance	Medical	Wheeled
Agustinos et al. (2014)	Vision	Contact w/humans	Medical	Arm
Bauzano et al. (2016)	Touch	Contact w/humans Remote guidance	Medical	Arm
Cortesao and Dominici (2017)	Touch	Contact w/humans	Medical	Arm
Maeda et al. (2001), Suphi Erden and Tomiyama (2010), Suphi Erden and Maric (2011), and Ficuciello et al. (2013)	Touch	Direct guidance	Production	Arm
Wang et al. (2015)	Touch	Carrying	Production	Wheeled
Bussy et al. (2012)	Touch	Carrying	Production	Humanoid
Baumeyer et al. (2015)	Touch	Remote guidance	Medical	Arm
Kumon et al. (2003, 2005), Magassouba et al. (2016b)	Audition	Contactless guidance	Service	Heads
Magassouba et al. (2015, 2016a,c)	Audition	Contactless guidance	Service	Wheeled
De Santis et al. (2007), Flacco et al. (2012), and Schlegl et al. (2013)	Distance	Collision avoidance	Production	Arm
Leboutet et al. (2016), Bergner et al. (2017), and Dean-Leon et al. (2017)	Distance	Collision avoidance	Service	Arm
Cherubini et al. (2016)	V+T (tra.)	Assembly	Production	Arm
Okuno et al. (2001), Okuno et al. (2004), and Hornstein et al. (2006)	V+A(tra.)	Contactless guidance	Service	Heads
Chan et al. (2012)	V+A(tra.)	Contactless guidance	Service	Wheeled
Papageorgiou et al. (2014)	V+T+A+D (tra.)	Direct guidance	Medical	Wheeled
Navarro et al. (2014)	D+T(tra.)	Collision avoidance	Production	Arm
Huang et al. (1999)	D+A(tra.)	Collision avoidance	Service	Wheeled
Natale et al. (2002)	V+A(sh.)	Contactless guidance	Service	Heads
Pomares et al. (2011)	V+T(hyb.)	Collision avoidance	Production	Arm
Chatelain et al. (2017)	V+T (hyb.)	Contact w/humans Remote guidance	Medical	Arm
Agravante et al. (2013, 2014)	V+T (sh.+hyb.)	Contact w/humans	Production	Humanoid
Cherubini and Chaumette (2013), Cherubini et al. (2014)	D+V (sh.+hyb.)	Collision avoidance	Production	Wheeled
Dean-Leon et al. (2016)	D+T (sh.+tra.)	Direct guidance	Service	Arm
Cherubini et al. (2015)	V+T (sh.+hyb.)	Assembly	Production	Arm

10.3389/fnbot.2020.576846

TABLE 4 | Classification based on target/potential sectors.

Production (manufacturing, transportation, construction)	Touch (Maeda et al., 2001; Suphi Erden and Tomiyama, 2010; Suphi Erden and Maric, 2011; Bussy et al., 2012; Ficuciello et al., 2013; Wang et al., 2015), distance (De Santis et al., 2007; Flacco et al., 2012; Schlegl et al., 2013), D+T (Navarro et al., 2014) V+T (Pomares et al., 2011; Agravante et al., 2013, 2014; Cherubini et al., 2015, 2016), V+D (Cherubini and Chaumette, 2013; Cherubini et al., 2014)
Medical (surgery, diagnosis, assistance)	Vision (Dune et al., 2008; Tsui et al., 2011; Agustinos et al., 2014; Narayanan et al., 2016), touch (Baumeyer et al., 2015; Bauzano et al., 2016; Cortesao and Dominici, 2017), V+T+A+D (Papageorgiou et al., 2014), V+T (Chatelain et al., 2017)
Service (companionship, domestic, personal)	Vision (Gridseth et al., 2015, 2016; Cai et al., 2016), audition (Kumon et al., 2005; Youssef et al., 2012; Magassouba et al., 2015, 2016a,b,c), distance (Leboutet et al., 2016; Bergner et al., 2017; Dean-Leon et al., 2017), V+A (Okuno et al., 2001, 2004; Natale et al., 2002; Hornstein et al., 2006; Chan et al., 2012), D+A (Huang et al., 1999), T+D (Dean-Leon et al., 2016)

TABLE 5 | Classification based on the type of robot platform.

Arms	Vision (Agustinos et al., 2014; Gridseth et al., 2015, 2016; Cai et al., 2016), touch (Maeda et al., 2001; Suphi Erden and Tomiyama, 2010; Suphi Erden and Maric, 2011; Ficuciello et al., 2013; Baumeyer et al., 2015; Bauzano et al., 2016; Cortesao and Dominici, 2017), distance (De Santis et al., 2007; Flacco et al., 2012; Schlegl et al., 2013; Leboutet et al., 2016; Bergner et al., 2017; Dean-Leon et al., 2017), V+T (Pomares et al., 2011; Cherubini et al., 2015, 2016; Chatelain et al., 2017), D+T (Navarro et al., 2014; Dean-Leon et al., 2016)
Wheeled	Vision (Dune et al., 2008; Tsui et al., 2011; Narayanan et al., 2016), touch (Wang et al., 2015), audition (Magassouba et al., 2015, 2016a,b), V+A (Chan et al., 2012), V+T+A+D (Papageorgiou et al., 2014), D+A (Huang et al., 1999), V+D (Cherubini and Chaumette, 2013; Cherubini et al., 2014)
Humanoids	Touch (Bussy et al., 2012), V+T (Agravante et al., 2013, 2014)
Heads	Audition (Kumon et al., 2003, 2005; Magassouba et al., 2016b), V+A (Okuno et al., 2001, 2004; Natale et al., 2002; Hornstein et al., 2006)

TABLE 3 | Classification based on the control objective with corresponding pHRI layer as proposed in De Luca and Flacco (2012) (in parenthesis).

Collision avoidance (*safety*)	Distance (De Santis et al., 2007; Flacco et al., 2012; Schlegl et al., 2013; Leboutet et al., 2016; Bergner et al., 2017; Dean-Leon et al., 2017), distance+touch (Navarro et al., 2014), Distance+audition (Huang et al., 1999), vision+touch (Pomares et al., 2011), Vision+distance (Cherubini and Chaumette, 2013; Cherubini et al., 2014)
Contact with passive humans (*coexistence*)	Vision (Agustinos et al., 2014), touch (Bauzano et al., 2016; Cortesao and Dominici, 2017), Vision+touch (Chatelain et al., 2017)
Contactless guidance (*collaboration*)	Vision (Dune et al., 2008; Tsui et al., 2011; Cai et al., 2016; Gridseth et al., 2016; Narayanan et al., 2016) Audition (Kumon et al., 2005; Youssef et al., 2012; Magassouba et al., 2015, 2016a,b,c) Vision+audition (Okuno et al., 2001, 2004; Natale et al., 2002; Hornstein et al., 2006; Chan et al., 2012)
Direct guidance (*collaboration*)	Touch+audition+distance+vision (Papageorgiou et al., 2014), Touch (Maeda et al., 2001; Suphi Erden and Tomiyama, 2010; Suphi Erden and Maric, 2011; Ficuciello et al., 2013), touch+distance (Dean-Leon et al., 2016)
Remote guidance (*collaboration*)	Vision (Agustinos et al., 2014; Gridseth et al., 2015), touch (Baumeyer et al., 2015; Bauzano et al., 2016), Vision+touch (Chatelain et al., 2017)
Collaborative assembly (*collaboration*)	Vision+touch (Cherubini et al., 2015, 2016)
Collaborative carrying (*collaboration*)	Touch (Bussy et al., 2012; Wang et al., 2015), vision+touch (Agravante et al., 2013, 2014)

10.3389/fnbot.2020.576846

TABLE 2 | Classification based on the sensors.

Vision	Dune et al., 2008; Tsui et al., 2011; Agustinos et al., 2014; Gridseth et al., 2015, 2016; Cai et al., 2016; Narayanan et al., 2016			
Touch	Maeda et al., 2001; Suphi Erden and Tomiyama, 2010; Suphi Erden and Maric, 2011; Bussy et al., 2012; Ficuciello et al., 2013; Baumeyer et al., 2015; Wang et al., 2015; Bauzano et al., 2016; Cortesao and Dominici, 2017	tra. (Cherubini et al., 2016), hyb. (Pomares et al., 2011; Chatelain et al., 2017) sh.+hyb. (Agravante et al., 2013, 2014; Cherubini et al., 2015)		
Audition	Kumon et al. (2003, 2005), Youssef et al. (2012), Magassouba et al. (2015, 2016a,b,c)	tra. Okuno et al. (2001, 2004), Hornstein et al. (2006), Chan et al. (2012), Papageorgiou et al. (2014), sh. (Natale et al., 2002)	tra. (Papageorgiou et al., 2014)	
Distance	De Santis et al., 2007; Flacco et al., 2012; Schlegl et al., 2013; Leboutet et al., 2016; Bergner et al., 2017; Dean-Leon et al., 2017	sh.+hyb. (Cherubini and Chaumette, 2013; Cherubini et al., 2014)	sh.+tra. (Dean-Leon et al., 2016)	tra. (Huang et al., 1999; Papageorgiou et al., 2014)
			tra. (Navarro et al., 2014)	
	Mono	Vision	Touch	Audition

Differentiate Robots & Mechanisms

The ability to adapt to changes of their subjects of operation or of their operating environment

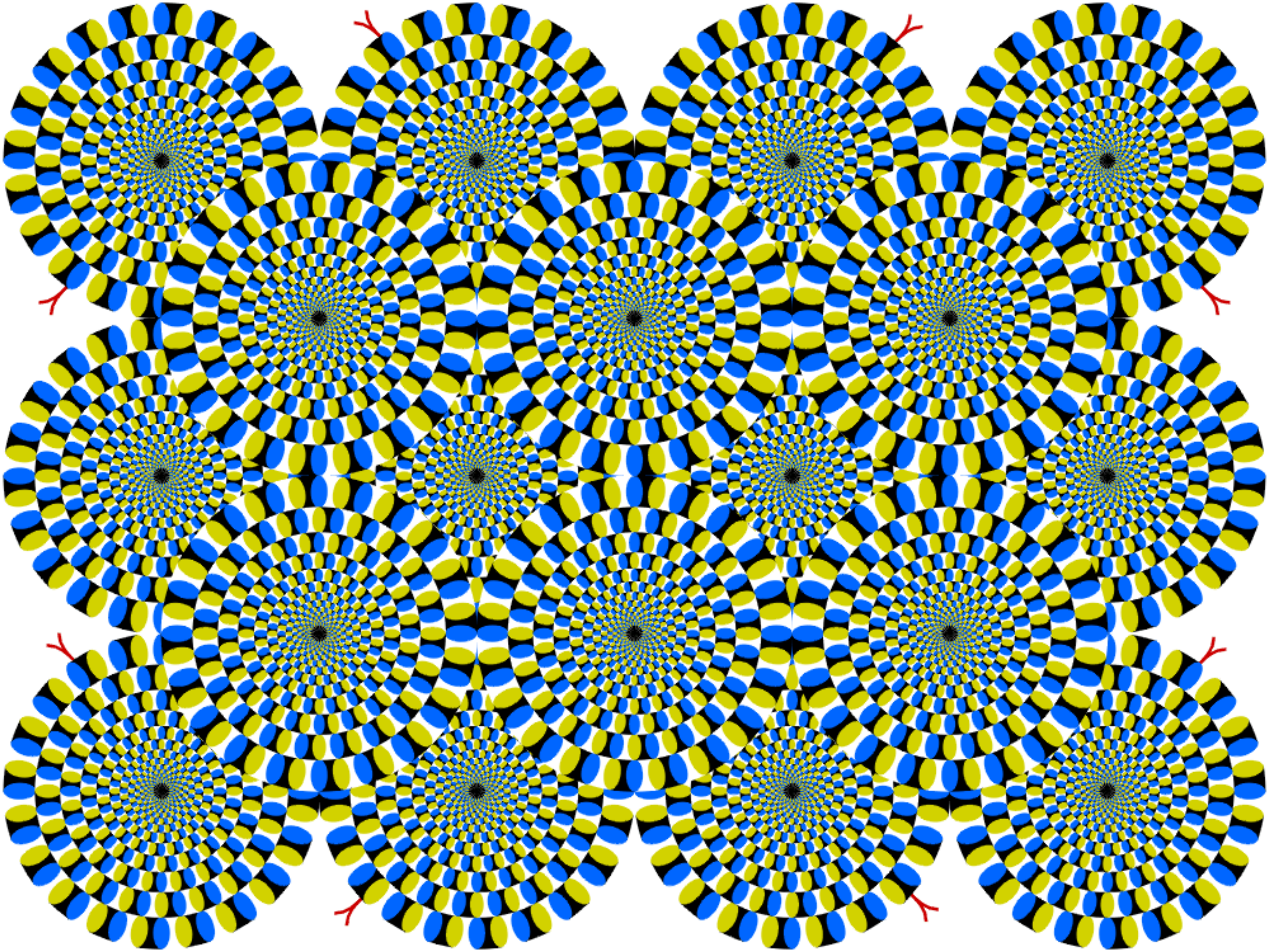
- To understand the surrounding environment
- To derive a set of actions from a high-level goal
- To implement (actuate and control) these actions

Robot Perception

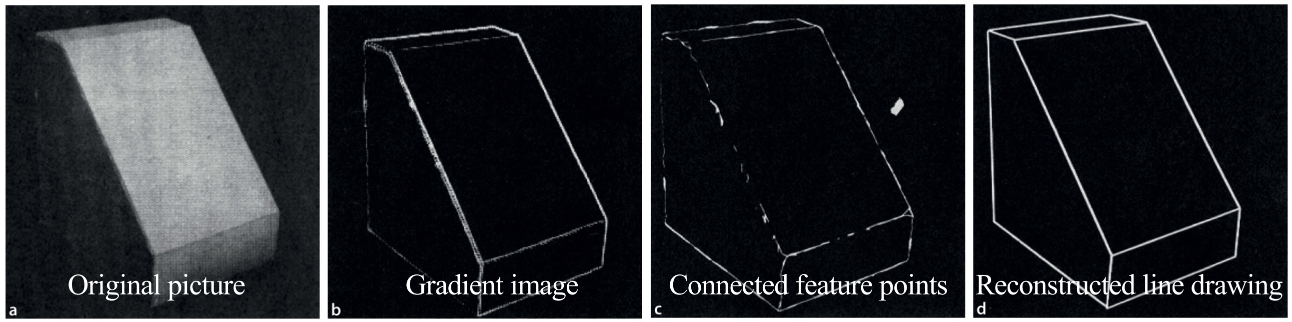
Physically implemented by sensors and by dedicated processing of the data they produce

- <u>To understand the surrounding environment</u>
- To derive a set of actions from a high-level goal
- <u>To implement (actuate and control) these actions</u>

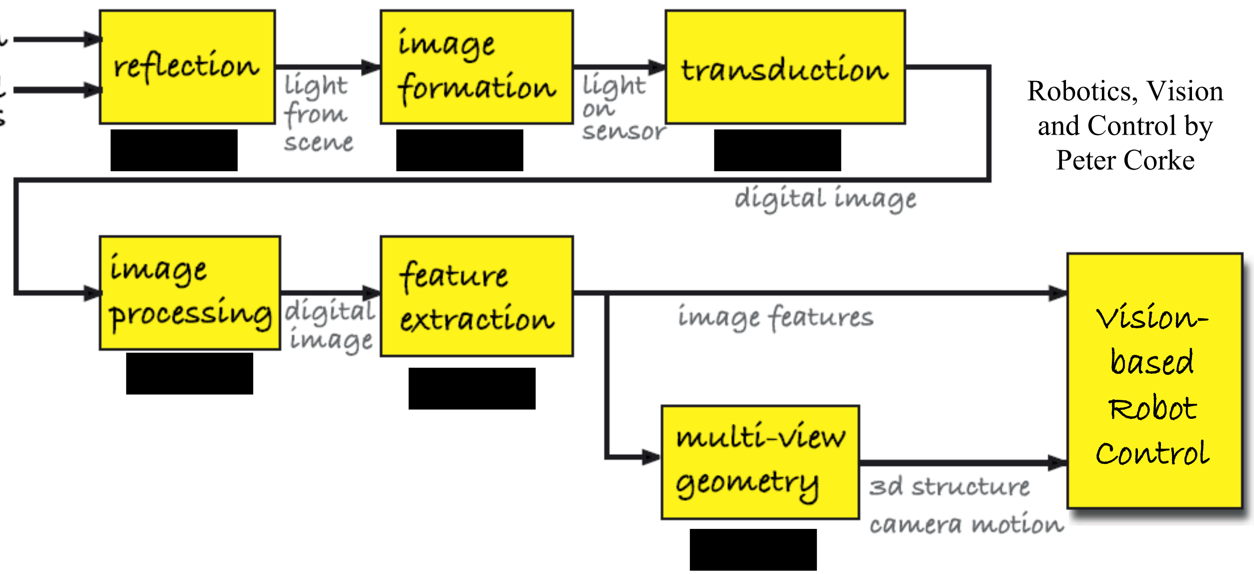
Why do robots need to see?



From Animals, to Computers, then Robots



Early results in computer vision for estimating the shape and pose of objects,
from the PhD work of L. G. Roberts at MIT Lincoln Lab in 1963.

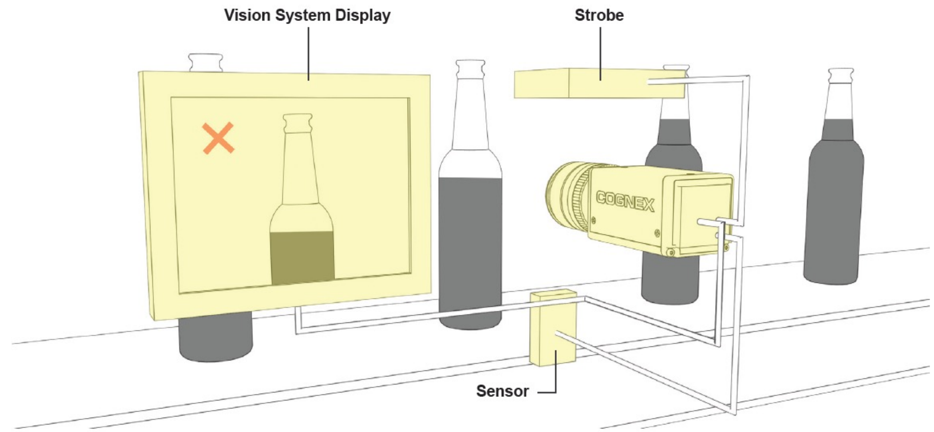


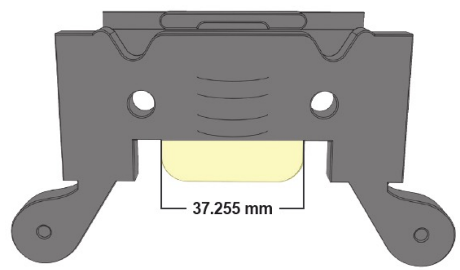
Robotics, Vision and Control by Peter Corke

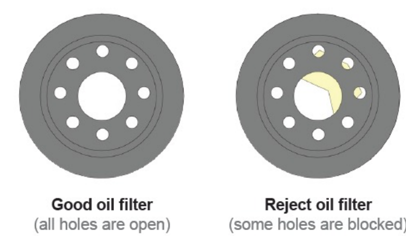
What Does Vision Tell Us About the World?

Or the Robot as a Machine

- Static features
  - Distance
  - Color
  - Shape
  - Texture
  - Environment
  - …







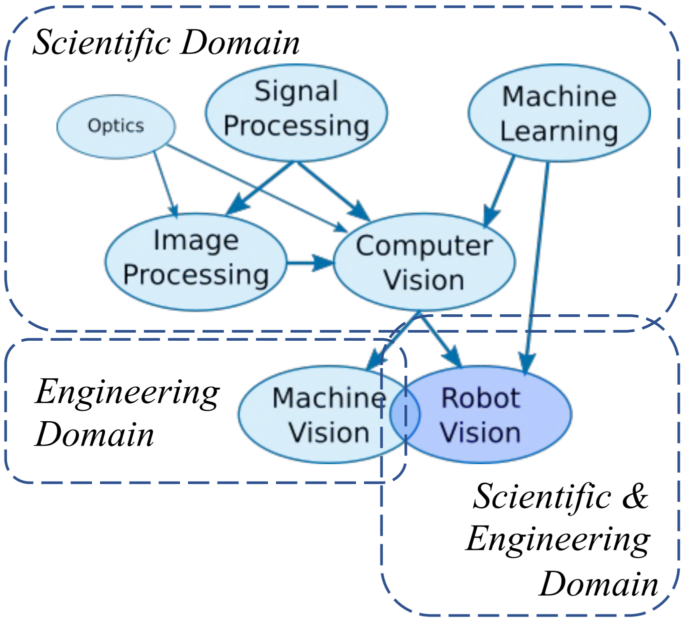
- Dynamic motions
- Understanding of the behavior
- An important method to interact with the physical world

Differentiating Concepts about Vision

- ***Signal Processing*** involves processing electronic signals to either clean them up, extract information, prepare them to output to a display or prepare them for further processing. Anything can be a signal, more or less.

- ***Computer Vision*** is more about extracting information from images to make sense of them.

- ***Image Processing*** techniques are primarily used to improve the quality of an image, convert it into another format (like a histogram) or otherwise change it for further processing.



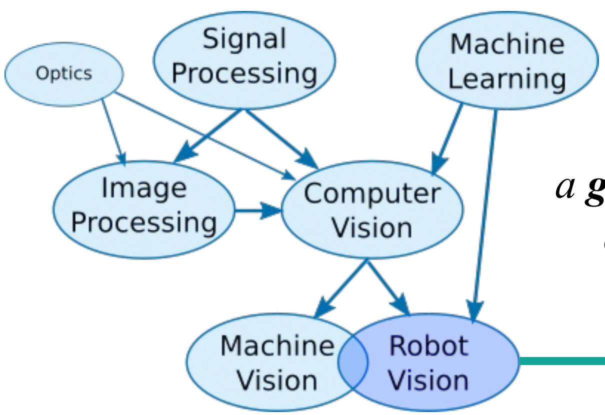
- ***Machine Learning*** is focused on recognizing patterns in data.

- ***Robotic Vision*** involves using a combination of camera hardware and computer algorithms to allow robots to process visual data from the world and execute physical actions.

- ***Machine Vision*** refers to the industrial use of vision for automatic inspection, process control and robot guidance.

Differentiating Concepts about 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**

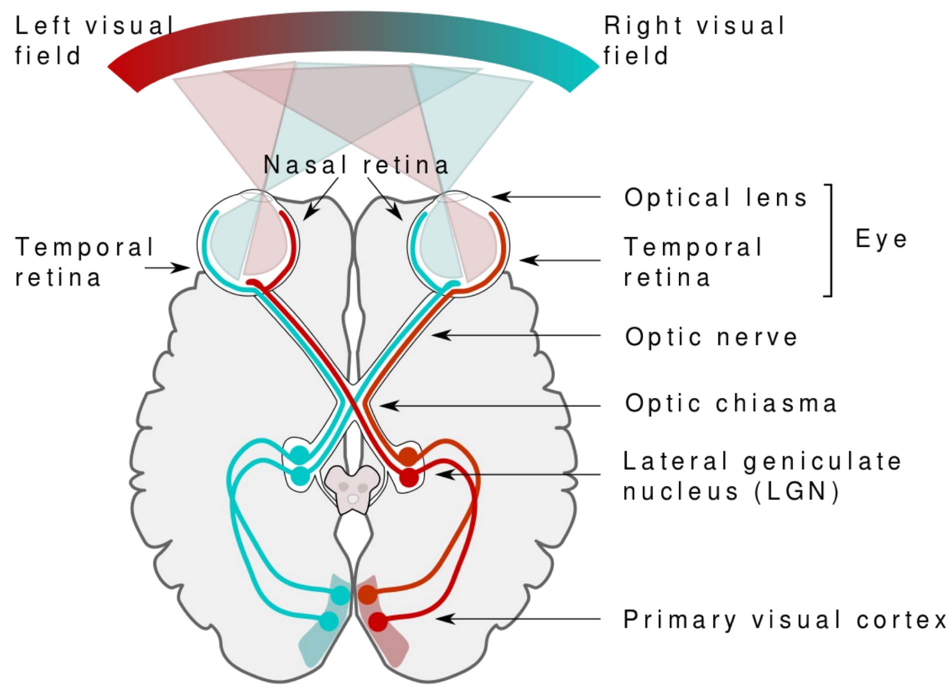


*a **goal** oriented **machine** that can **sense**, **plan** and **act***

What is Vision?

Vision System

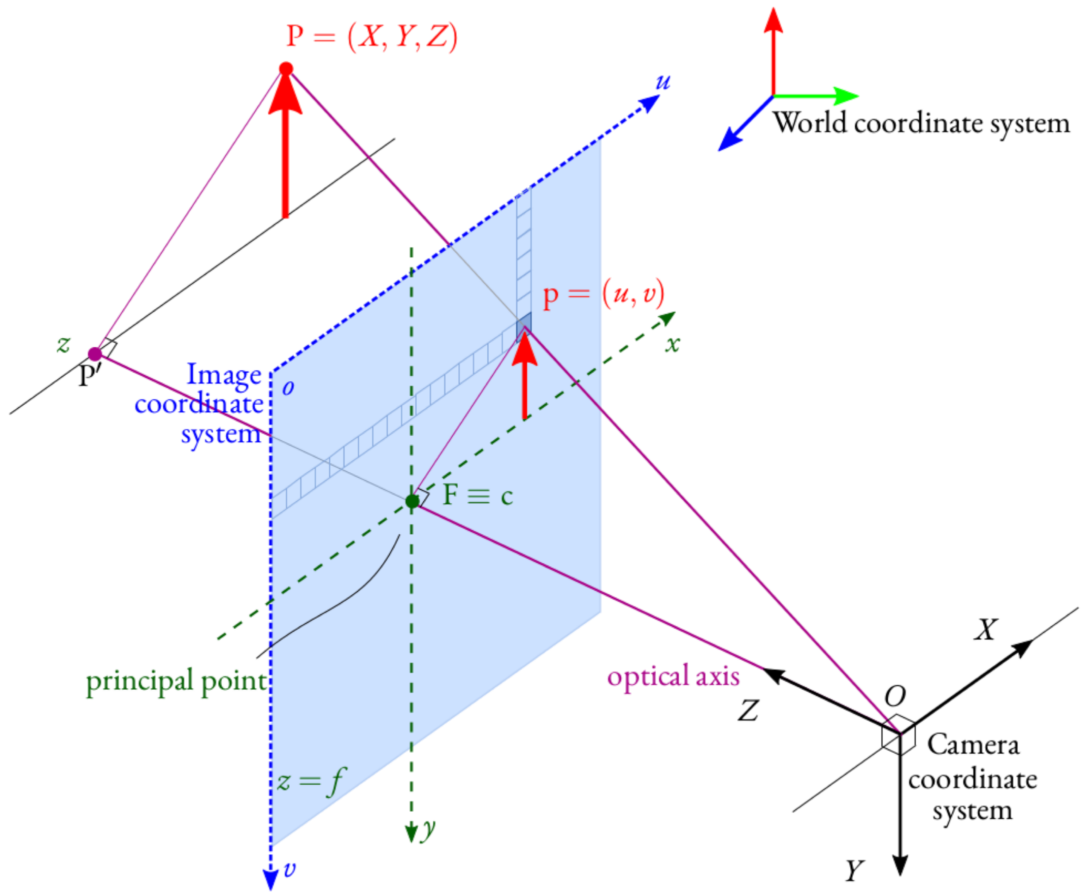
- Visual perception is the act of observing patterns and objects through sight
- Visual systems allow us to build a model of the physical world.



What is Vision?

Vision System

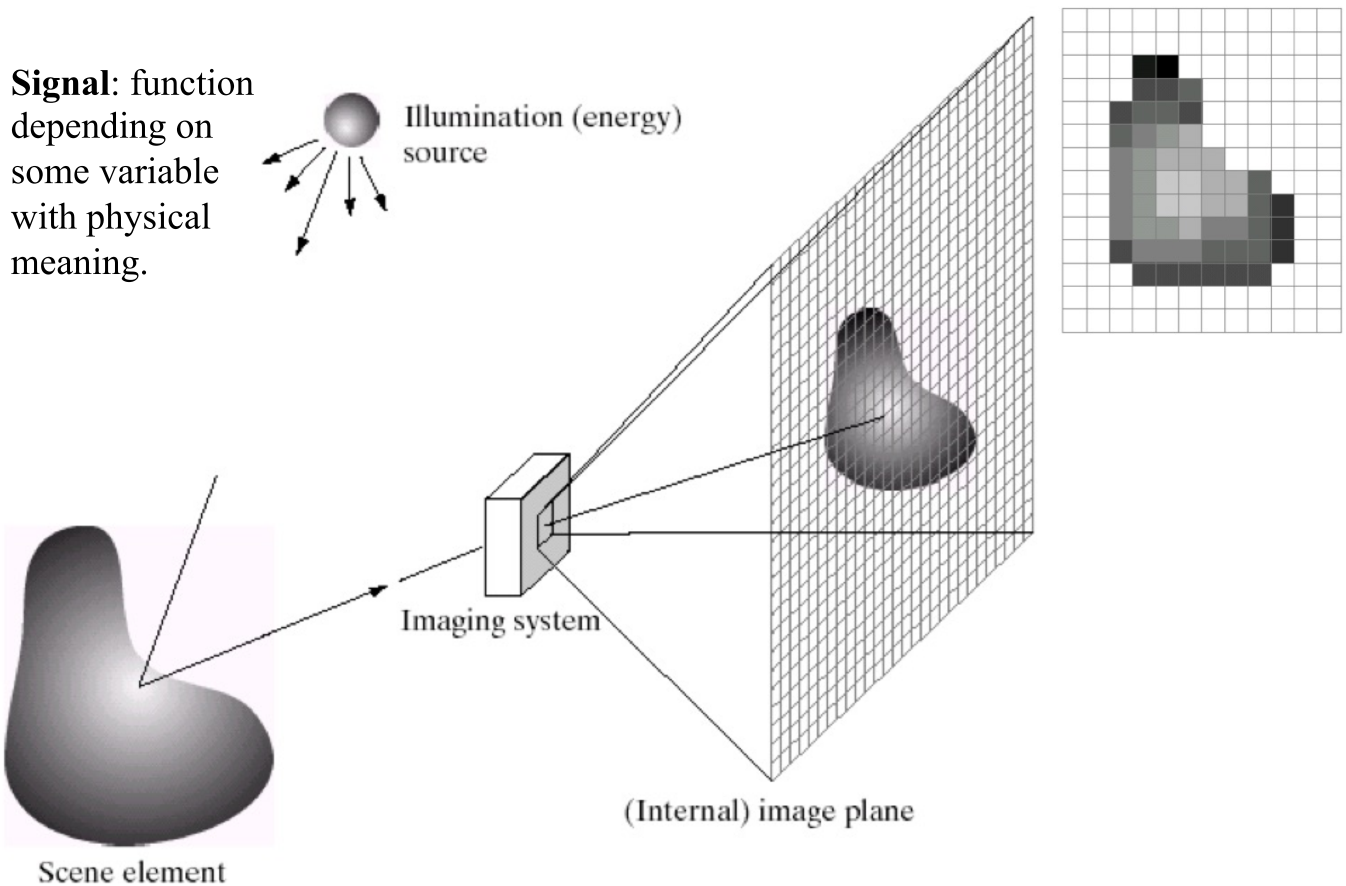
- Visual perception is the act of observing patterns and objects through sight
- Visual systems allow us to build a model of the physical world.



Formulation of An Image

As a 2D sampling of signal

Signal: function depending on some variable with physical meaning.

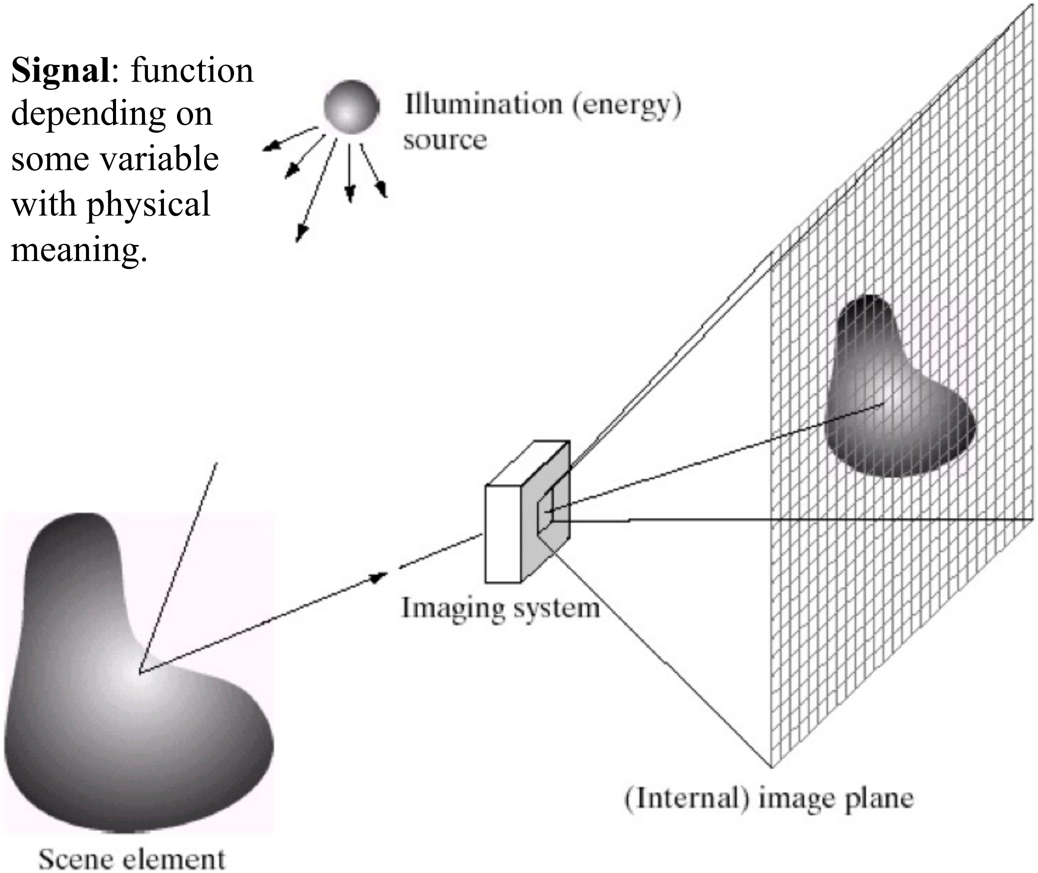


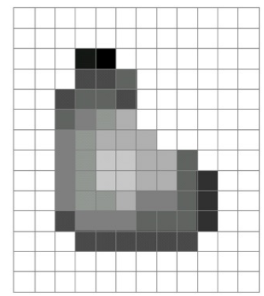
Formulation of An Image

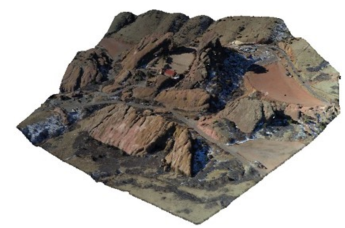
As a 2D sampling of signal

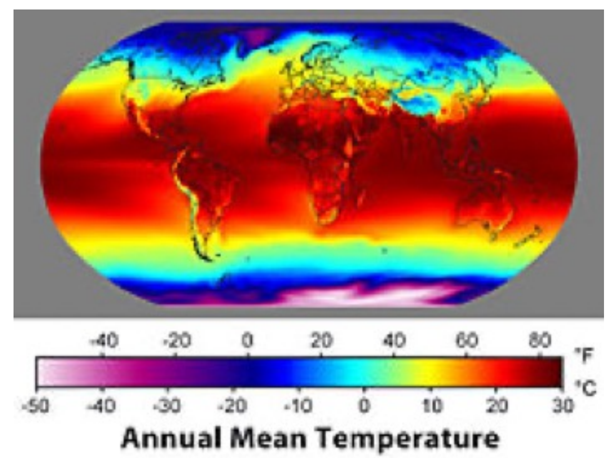
Can be other physical values too: *temperature, pressure, depth* …

Signal: function depending on some variable with physical meaning.





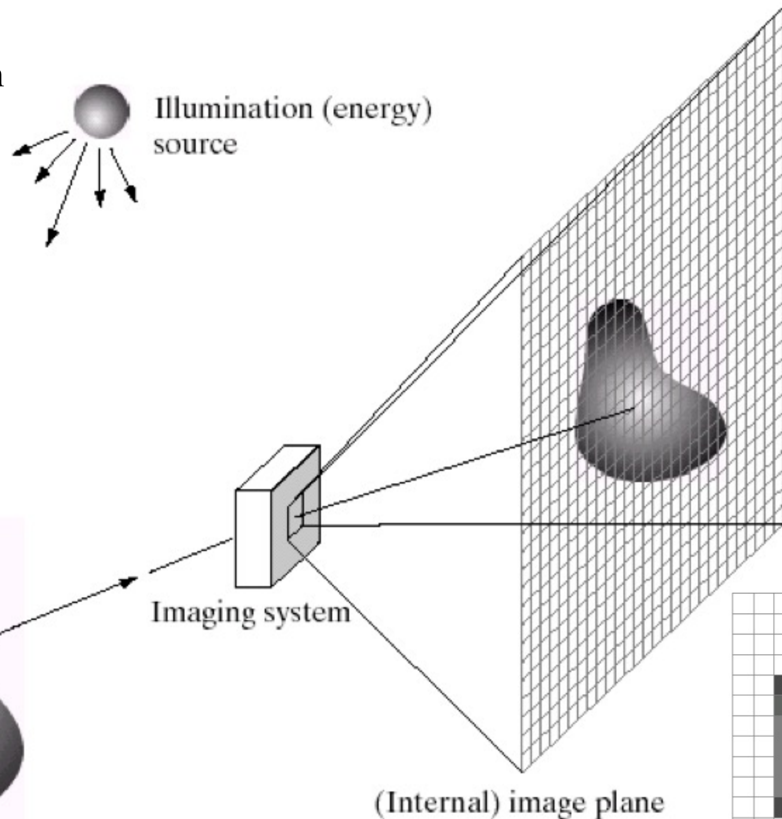




Formulation of An Image

As a 2D sampling of signal

Signal: function depending on some variable with physical meaning.



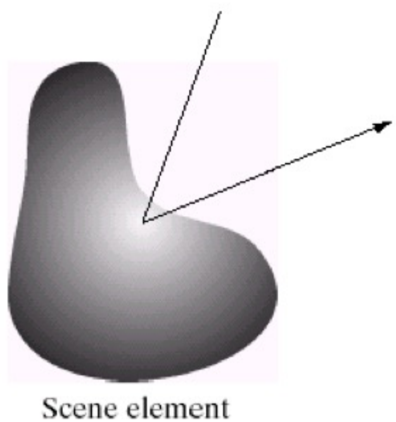
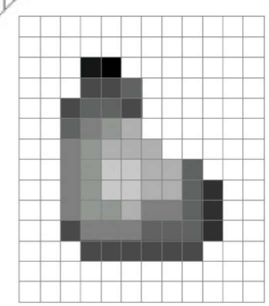
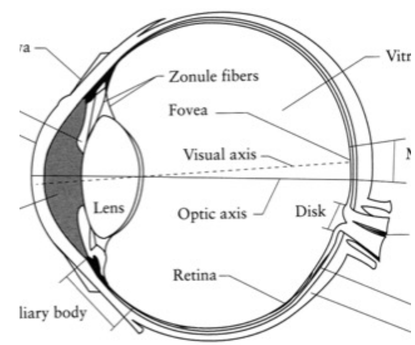


Image: sampling of that function.

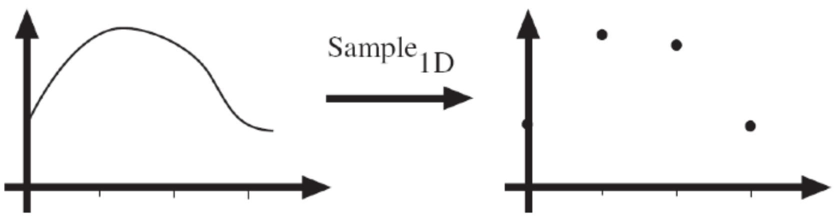
- 2 variables: xy coordinates
- 3 variables: xy + time (video)
- ‘Brightness’ is the value of the function for visible light





Sampling Physical World Using Images

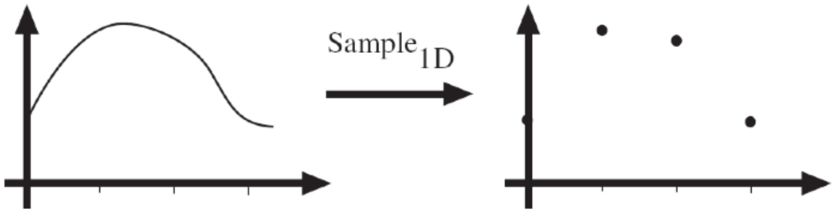
Physical Understanding of Images



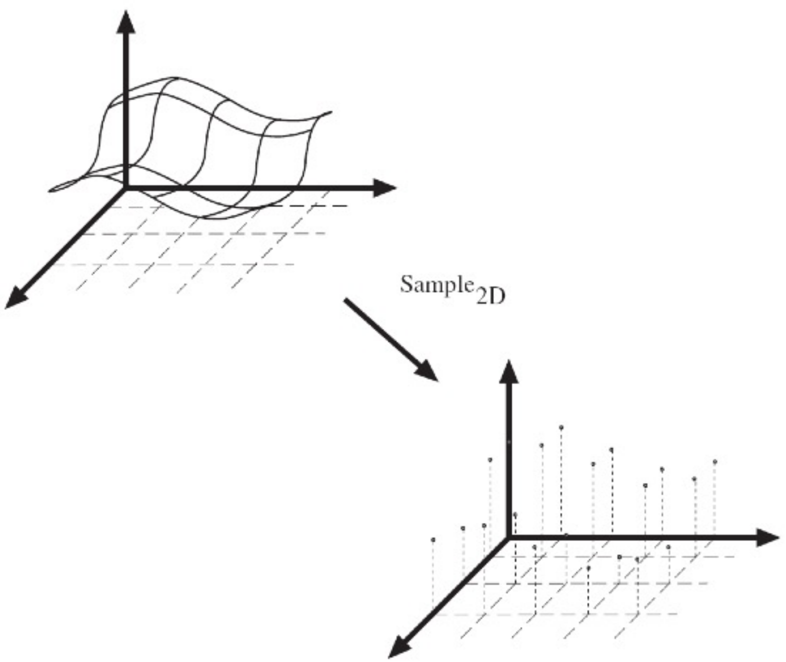
- Sampling in 1D takes a function and returns a vector whose elements are values of that function at the sample points.

Sampling Physical World Using Images

Physical Understanding of Images



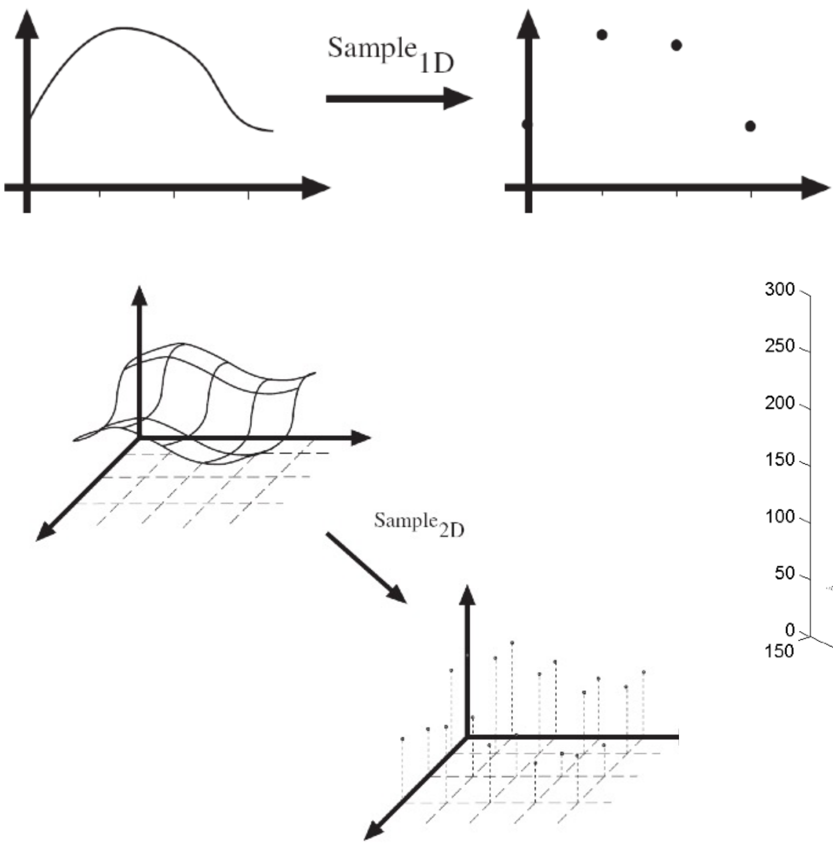
- Sampling in 1D takes a function and returns a vector whose elements are values of that function at the sample points.



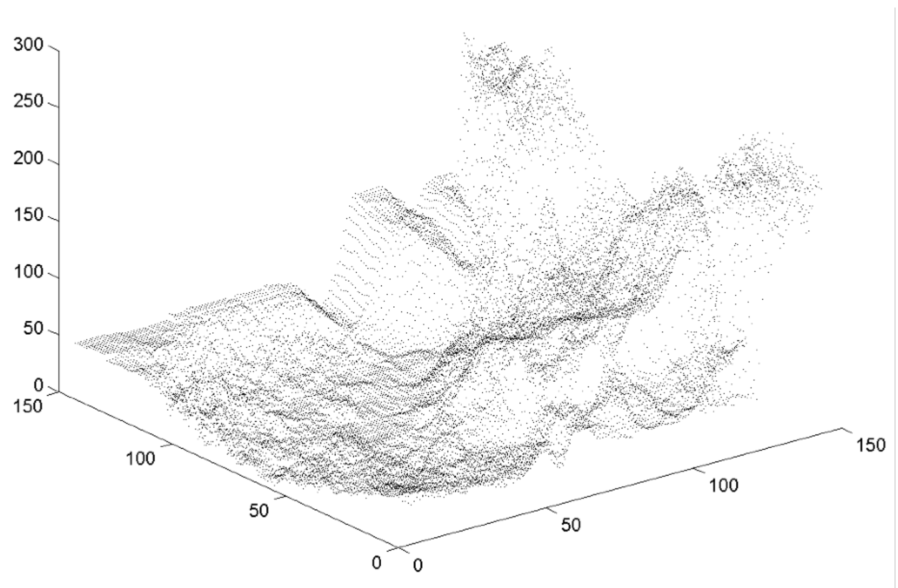
- Sampling in 2D takes a function and returns a matrix.

Sampling Physical World Using Images

Physical Understanding of Images



Grayscale Digital Image



Sampling Physical World Using High-framerate Depth Sensing

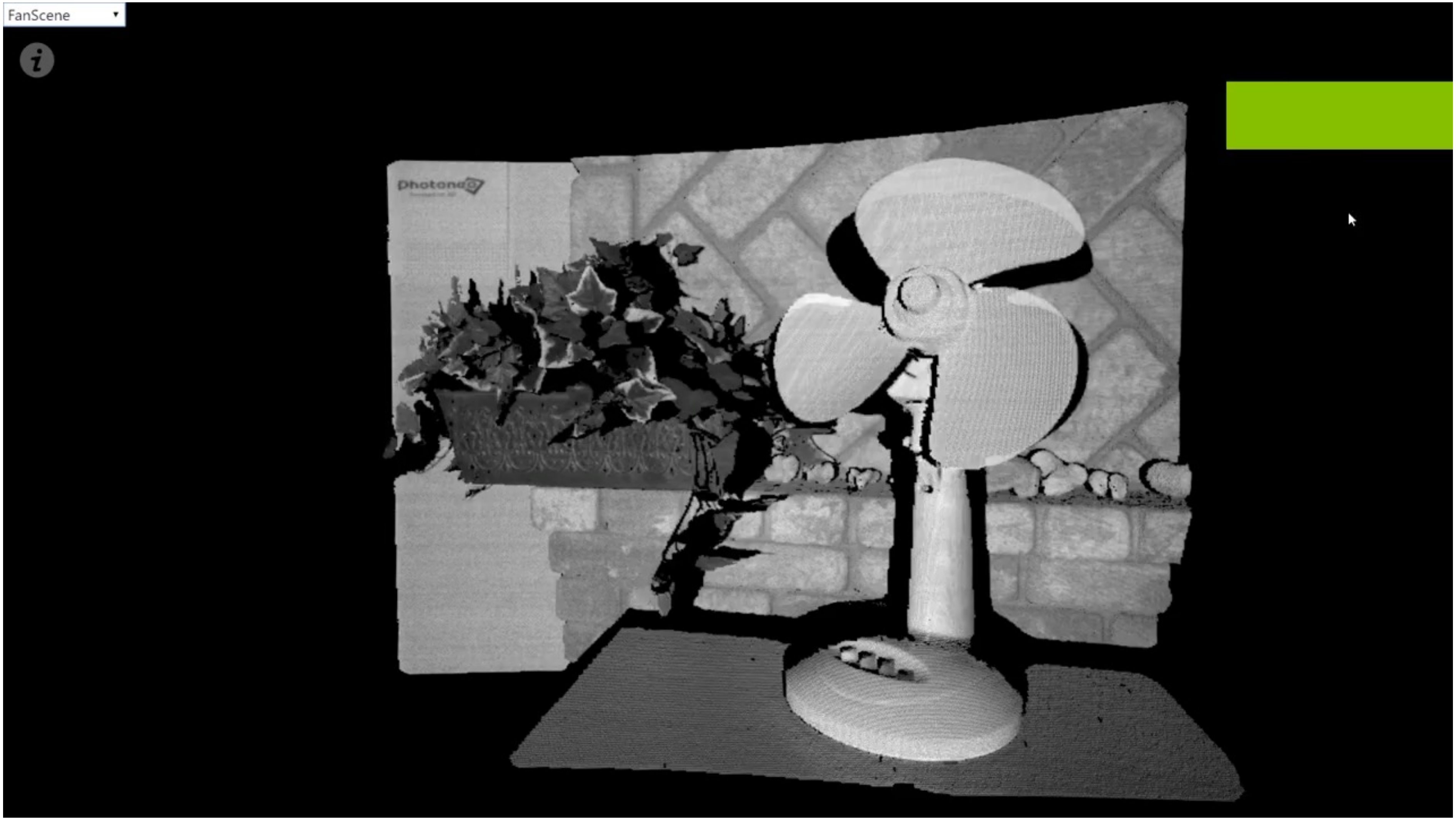


Image Representation

Example of a grayscale [0, 1] image within a planar area of size [m, n]

In [1]:

```
import numpy as np
from numpy import random as r
```

In [2]:

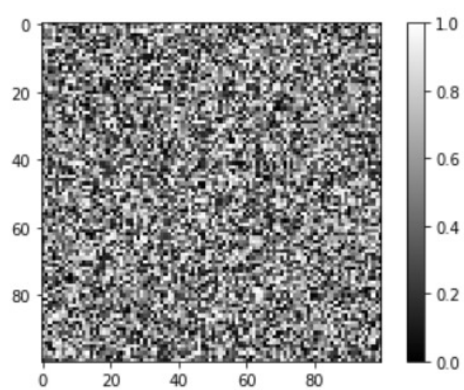
```
from matplotlib import pyplot as p
I = r.rand(100,100);
```

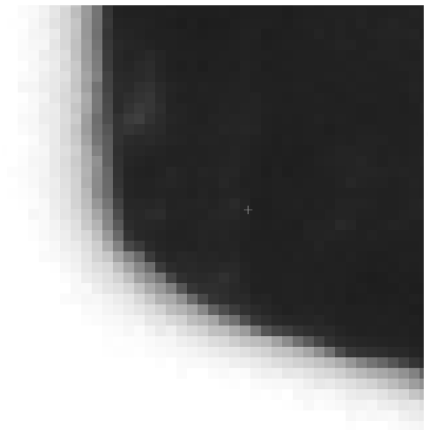
In [3]:

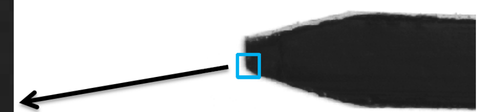
```
p.imshow(I, cmap="gray", vmin=0.0, vmax=1.0);
p.colorbar()
I[0,1]
```

Out[3]:

```
0.9564898647579192
```

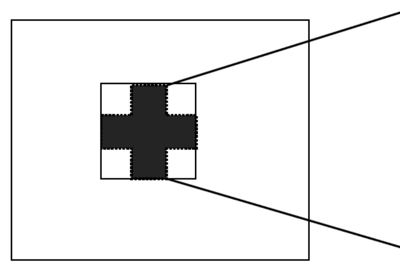






Pixel as picture element

NOT A SQUARE !!!

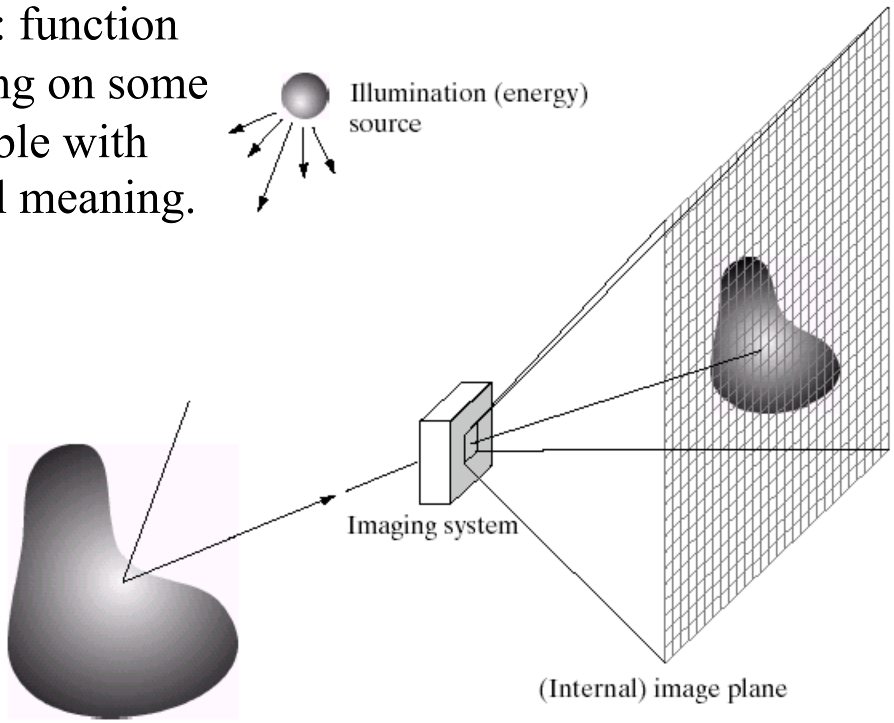


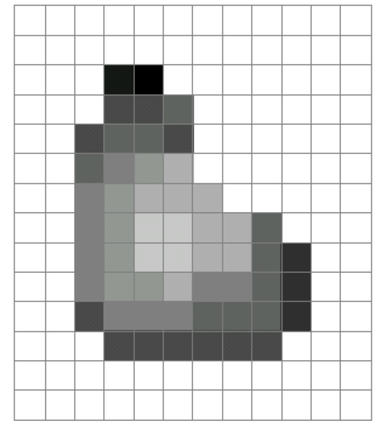
255	255	255	105	51	41	43	49	101	255	255	255
255	255	255	116	62	44	42	57	120	255	255	255
255	255	255	112	68	41	46	58	117	255	255	255
105	110	111	109	60	42	48	61	115	112	114	108
60	68	62	57	42	41	46	41	43	49	42	41
44	42	41	46	46	42	48	44	42	42	46	42
41	46	42	48	44	42	41	41	46	43	49	42
59	54	60	59	41	46	42	46	46	42	48	46
100	120	120	115	51	41	43	49	110	116	118	105
255	255	255	118	62	44	42	57	115	255	255	255
255	255	255	121	68	41	46	58	120	255	255	255
255	255	255	100	60	42	48	61	105	255	255	255

Formulation of An Image

As a 2D sampling of signal

Signal: function depending on some variable with physical meaning.

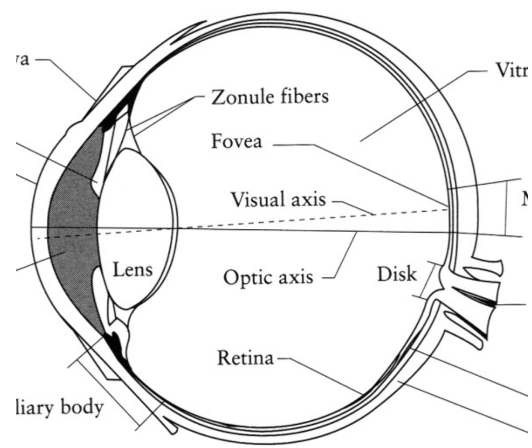




Can be other physical values too: *temperature, pressure, depth* …

Image: sampling of that function.

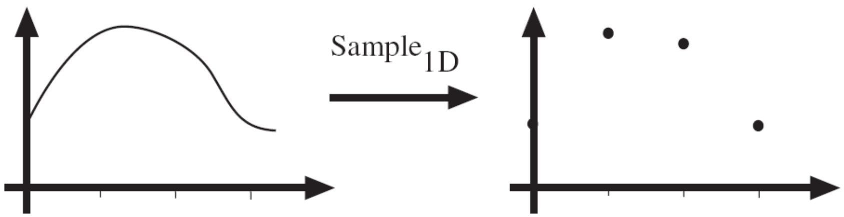
- 2 variables: xy coordinates
- 3 variables: xy + time (video)
- 'Brightness' is the value of the function for visible light



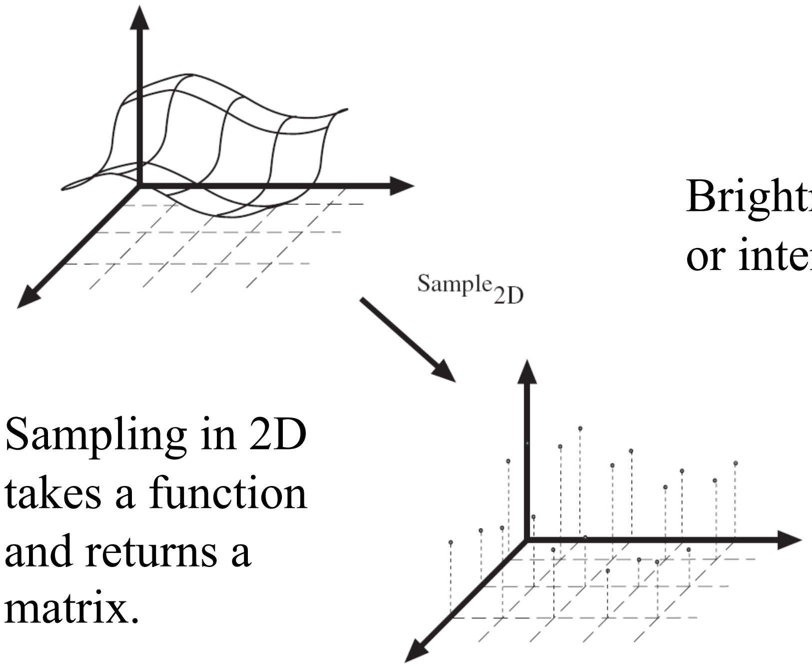
AncoraSIR.com

Sampling Physical World Using Images

Physical Understanding of Images

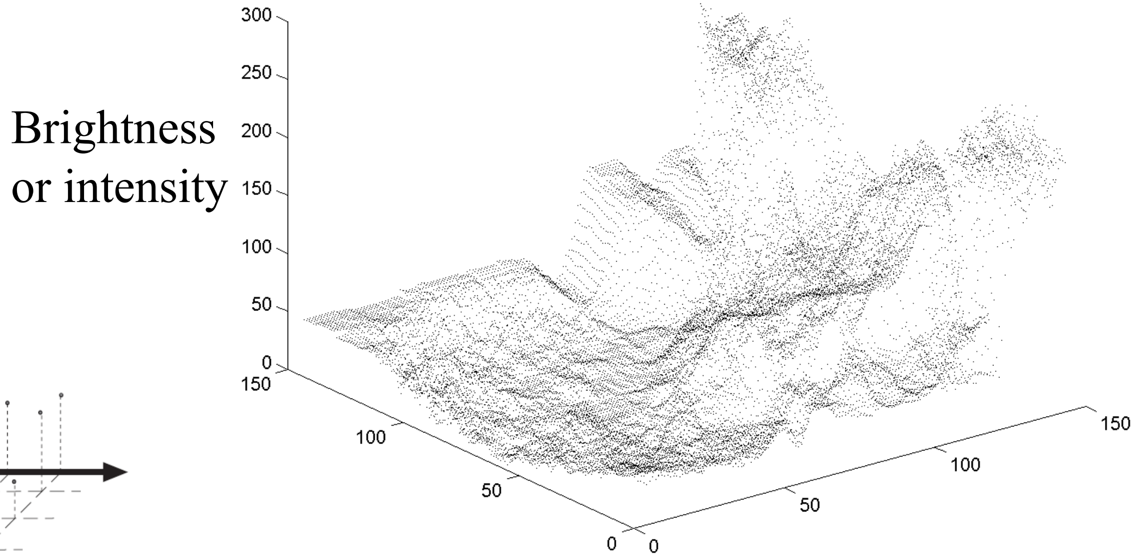


Sampling in 1D takes a function, and returns a vector whose elements are values of that function at the sample points.

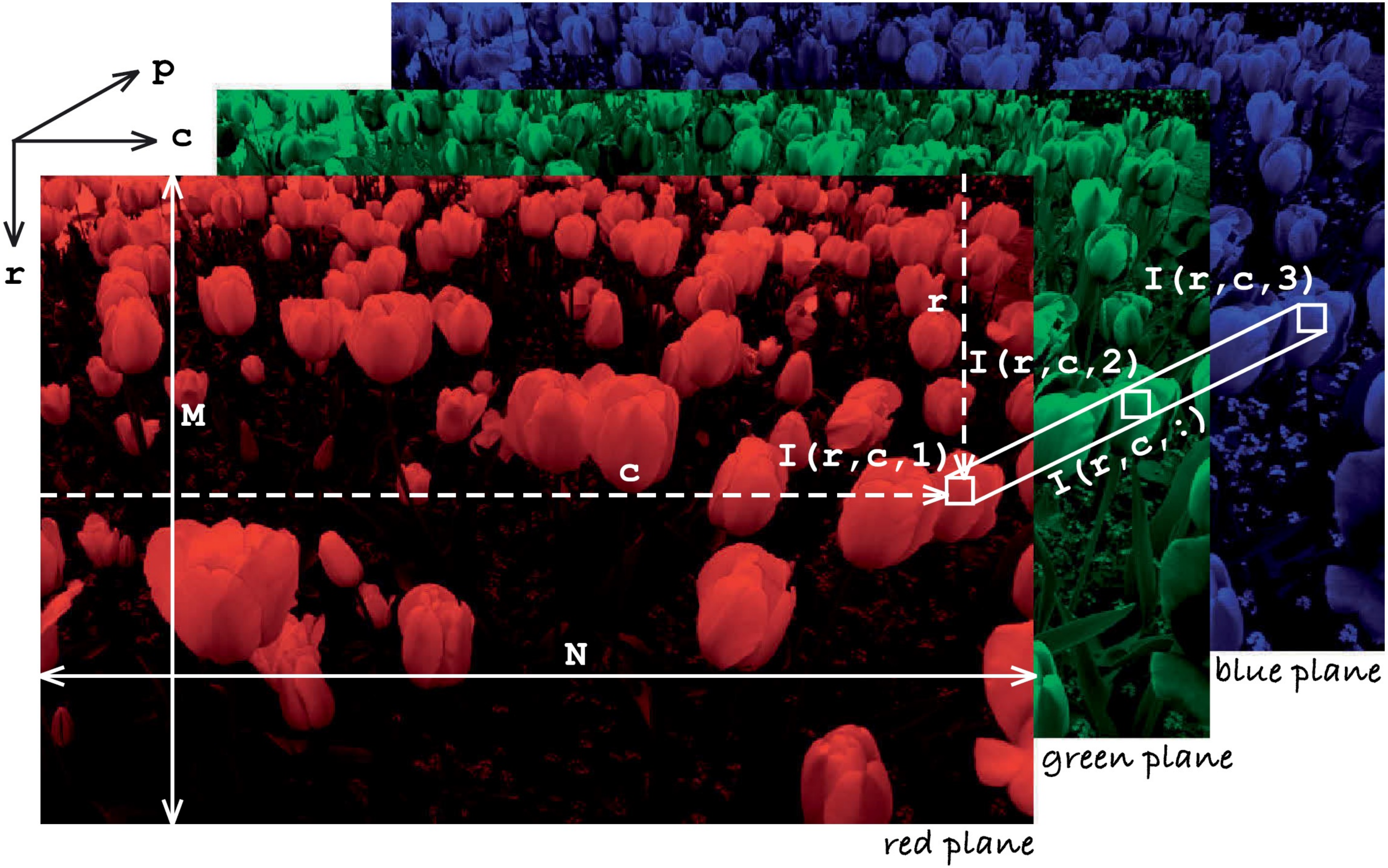


Sampling in 2D takes a function and returns a matrix.

Grayscale Digital Image



An RGB Image



A Robotic Way of Interpreting Images

An important method of sensing the environment

- Computer Vision
  - Digitization of physical world in multi-dimensional linear algebra
  - *Physical meaning is not a required way of interpretation or usage*
- Robotic Vision
  - Same as Computer Vision, but with a focus on Physical Interpretation
  - *Because actions need to be executed **by a robot** and people might get hurt*

Machine Vision *actually measures but no action required.*

Image Size: u, v

Time Series: t

Other variables
Heatmap: H
Temperature: T
…

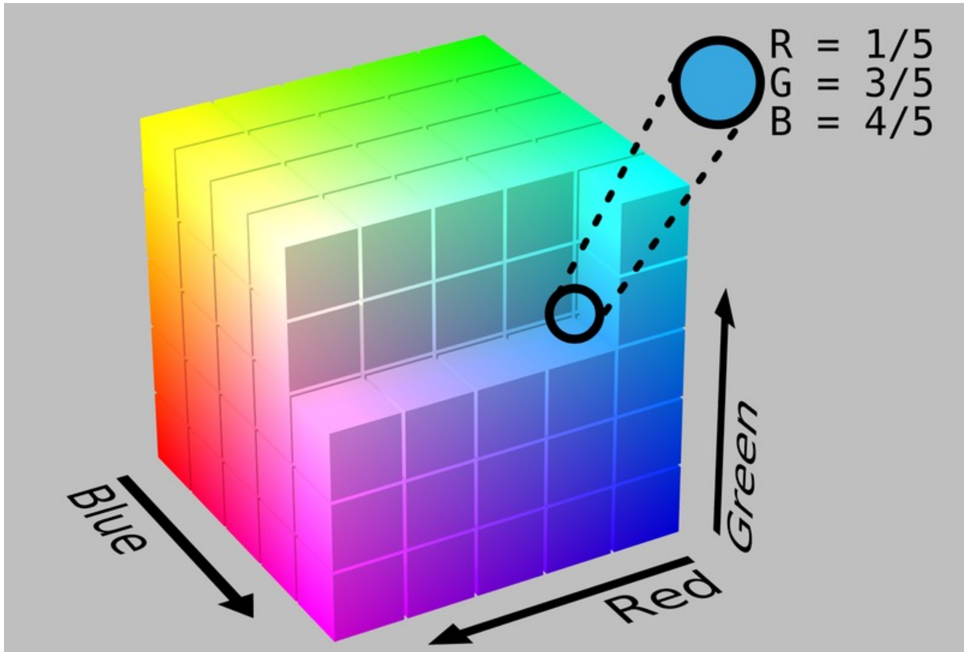
Color Space: **R**ed, **G**reen, **B**lue
Grayscale: **Gray**

Point Cloud: $x(u, v)$, $y(u, v)$, $z(u, v)$

Texture: $r(x, y, z)$, $g(x, y, z)$, $b(x, y, z)$

[0, 1] *as normalized form, not an integer*
[0, 255] *as a byte number of range* $2^8=256$ *from 0 to 255, all in integer forms*

AncoraSIR.com



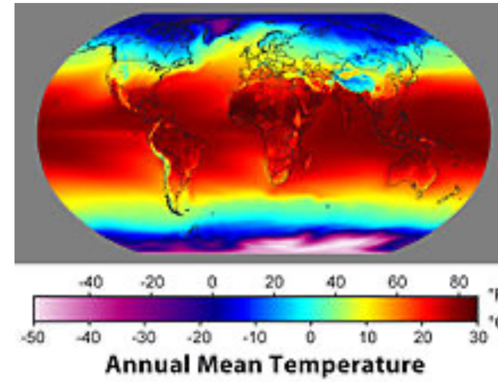
Color Space: **R**ed, **G**reen, **B**lue

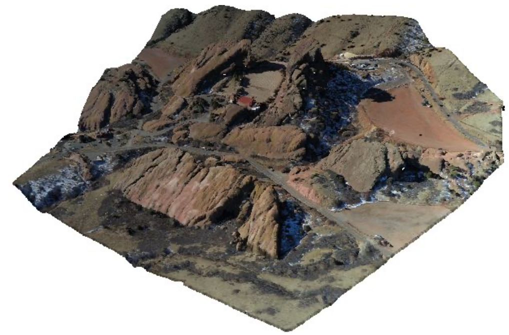
Other variables

Heatmap: H

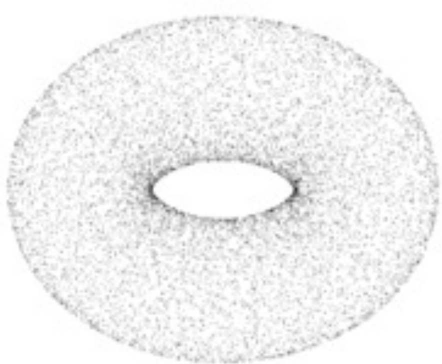
Temperature: T

…





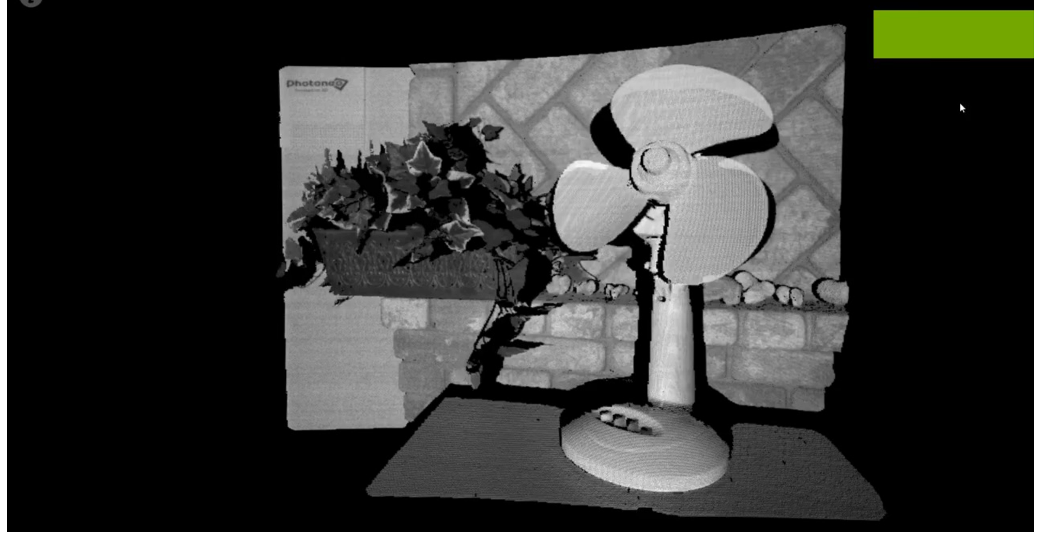
Texture:
$r(x, y, z)$,
$g(x, y, z)$,
$b(x, y, z)$



Point Cloud: $x(u, v), y(u, v), z(u, v)$

AncoraSIR.com

Time Series: t



Perspective Transform

Camera Models

Lens Law $\dfrac{1}{z_o} + \dfrac{1}{z_i} = \dfrac{1}{f}$

the focal length of the lens (f)

the distance to the object (z_o)

the distance to the image (z_i)

perspective projection $x = f\dfrac{X}{Z},\ y = f\dfrac{Y}{Z}$

$\boldsymbol{P} = (X, Y, Z) \xrightarrow{f} \boldsymbol{p} = (x, y)$

homogeneous form

$\tilde{\boldsymbol{p}} = (x', y', z')$

$x' = \dfrac{fX}{z'},\ y' = \dfrac{fY}{z'},\ z' = Z$

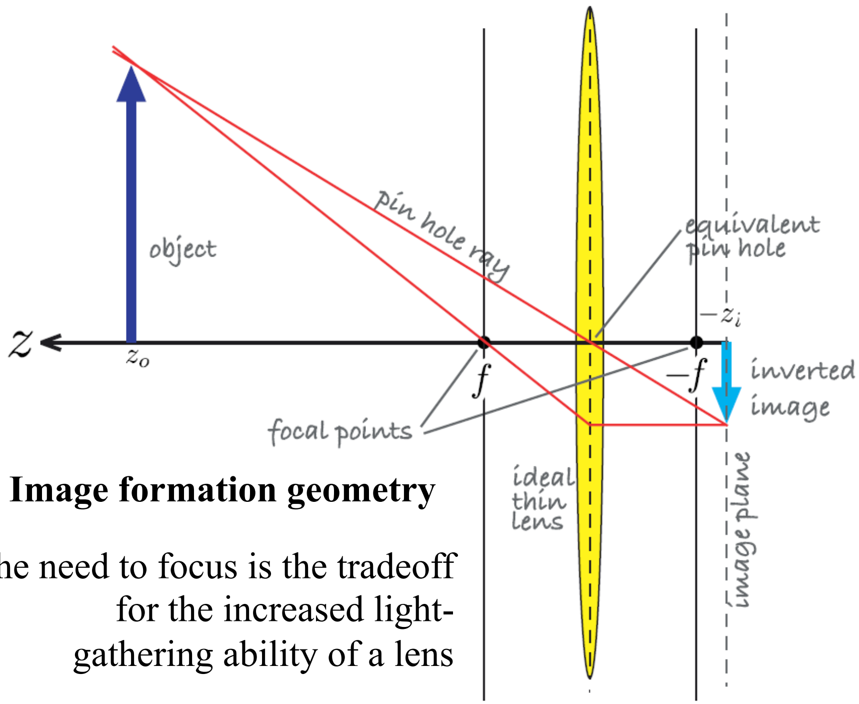
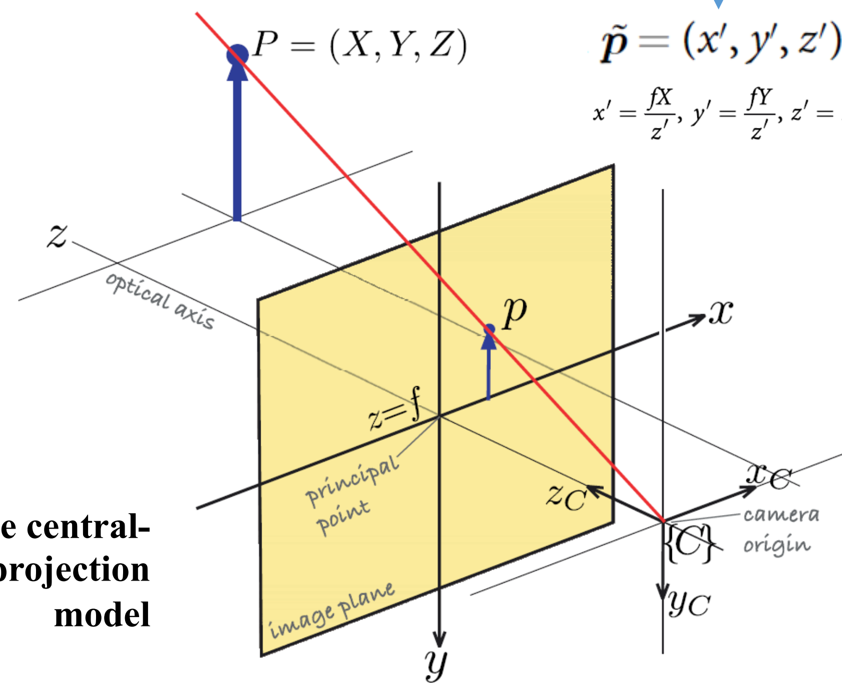


Image formation geometry

the need to focus is the tradeoff for the increased light-gathering ability of a lens



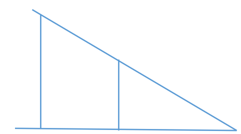
The central-projection model

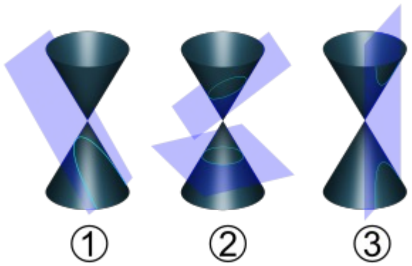
Characteristics

Perspective Transform

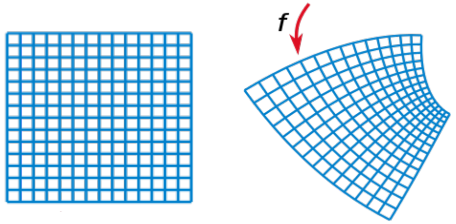
- A mapping from 3D space to the 2D image plane
- Straight lines in the world are projected to straight lines on the image plane.
- Parallel lines in the world are projected to lines that intersect at a vanishing point.
  - The exception are lines in the plane parallel to the image plane which do not converge.
- Conics in the world are projected to conics on the image plane.
- **The mapping is not one-to-one and a unique inverse does not exist.**
- The transformation is not conformal
  - It does not preserve shape since internal angles are not preserved, different from translation, rotation and scaling.

$$\mathbb{R}^3 \mapsto \mathbb{R}^2.$$





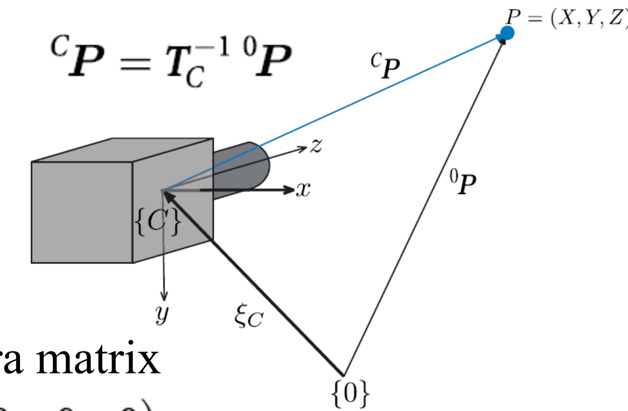
$$\boldsymbol{P} = (X, Y, Z) \xrightarrow{f} \boldsymbol{p} = (x, y)$$



Retinal Image Plane Coordinates

Written in homogeneous form

$${}^{C}\boldsymbol{P} = \boldsymbol{T}_C^{-1}\,{}^{0}\boldsymbol{P}$$



$$\boldsymbol{P} = (X, Y, Z) \xrightarrow{\text{homogeneous form}} {}^{C}\tilde{\boldsymbol{P}} = (X, Y, Z, 1)^T$$

$$x' = \frac{fX}{z'},\ y' = \frac{fY}{z'},\ z' = Z$$

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

$$\tilde{\boldsymbol{p}} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

camera matrix

$$\tilde{\boldsymbol{p}} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} {}^{C}\tilde{\boldsymbol{P}}$$

$$\tilde{\boldsymbol{p}} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} {}^{C}\tilde{\boldsymbol{P}}$$

$$\boldsymbol{p} = (x, y) \xrightarrow{\text{homogeneous form}} \tilde{\boldsymbol{p}} = (x', y', z')$$

$$x = \frac{x'}{z'},\ y = \frac{y'}{z'}$$

What about the physical coordinates on the actual image?

AncoraSIR.com

Express w.r.t the Camera

Physical Meanings of Camera Pixels

- A **camera sensor** with a **W**× **H** grid of image pixels
  - The pixel coordinates (u, v)

Principal point in pixel coordinate

$$u = \frac{x}{\rho_w} + u_0,\ v = \frac{y}{\rho_h} + v_0$$

width and height of each pixel

$$\tilde{\boldsymbol{p}} = (u', v', w')$$

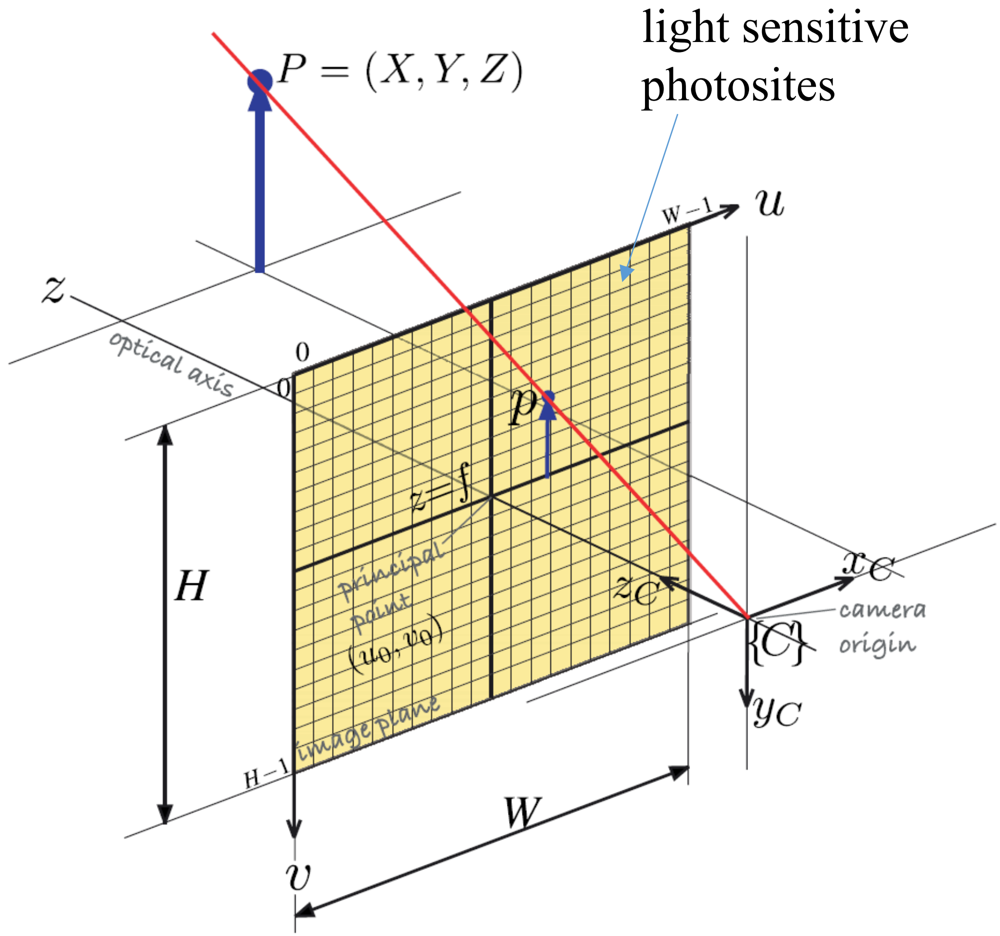
pixel coordinate

$$u = \frac{u'}{w'},\ v = \frac{v'}{w'}$$

camera (projection) matrix

$$\tilde{\boldsymbol{p}} = \underbrace{\begin{pmatrix} 1/\rho_w & 0 & u_0 \\ 0 & 1/\rho_h & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{K} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} {}^C\tilde{\boldsymbol{P}}$$

camera parameter matrix



AncoraSIR.com

Camera Projection In General Form

Still, something is missing

- The 3x4 Camera Calibration Matrix
  - Performs scaling, translation and perspective projection

$$\tilde{\boldsymbol{p}} = \underbrace{\begin{pmatrix} f/\rho_w & 0 & u_0 \\ 0 & f/\rho_h & v_0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{intrinsic}} \underbrace{\left({}^0\mathrm{T}_C\right)^{-1}}_{\text{extrinsic}} \tilde{\boldsymbol{P}}$$

$$= \boldsymbol{K}\boldsymbol{P}_0 \, {}^0\boldsymbol{T}_C^{-1}\tilde{\boldsymbol{P}}$$

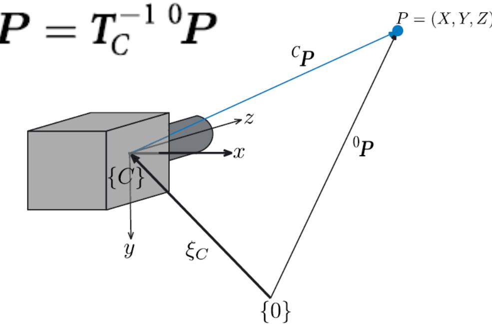
$$= \boldsymbol{C}\tilde{\boldsymbol{P}}$$

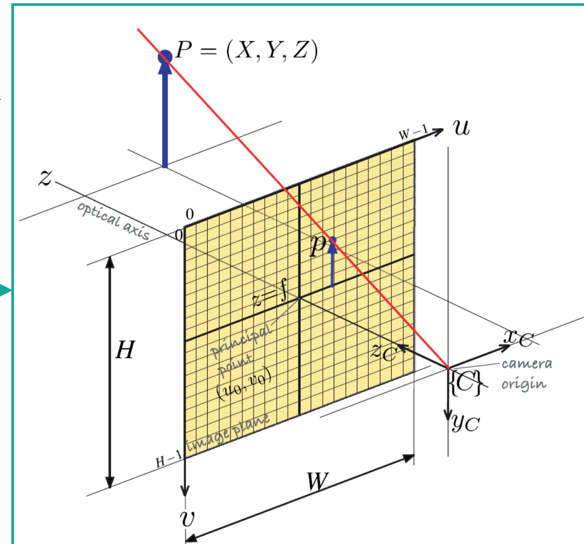
$$\tilde{\boldsymbol{p}} = (\boldsymbol{C}\boldsymbol{H}^{-1})(\boldsymbol{H}\tilde{\boldsymbol{P}}) = \boldsymbol{C}'\tilde{\boldsymbol{P}}'$$

C is 3x4 with **12** elements

Unconstrained Overall Scale Factor

$${}^C\boldsymbol{P} = \boldsymbol{T}_C^{-1}\,{}^0\boldsymbol{P}$$





$$\boldsymbol{p} = \mathcal{P}(\boxed{\boldsymbol{P}}, \boxed{\boldsymbol{K}}, \boxed{\xi_C})$$

Camera pose: **6** parameters

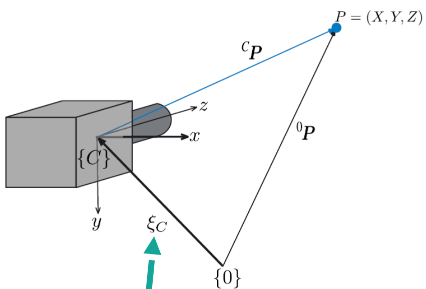
Camera parameter matrix: **5** parameters

It can only be solved if we have information about the camera or the 3D object

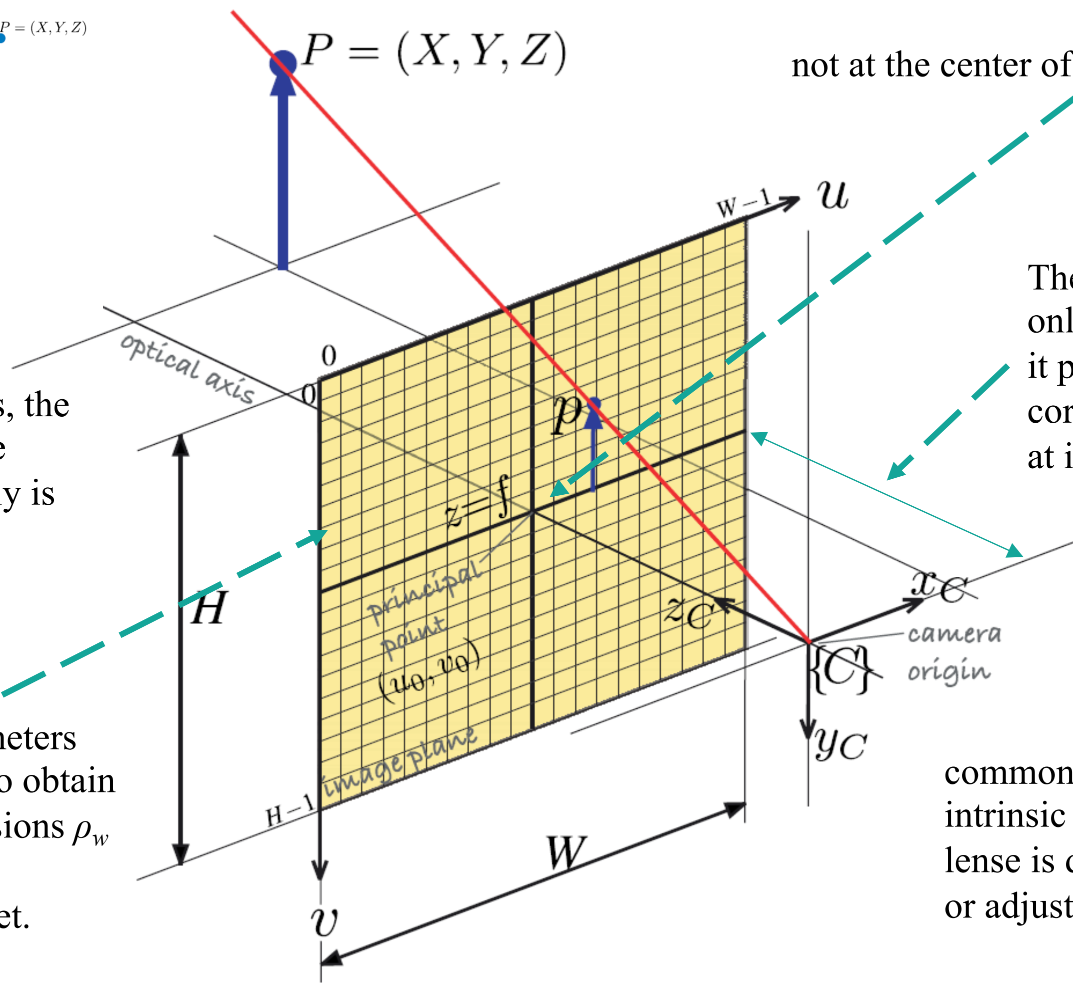
AncoraSIR.com

Camera Calibration

In general, the cameras are not made as modeled



The extrinsic parameters, the camera's pose, raises the question of where exactly is the center point of the camera.



not at the center of the photosite array

The focal length of a lens is only accurate to 4% of what it purports to be, and is only correct if the lens is focused at infinity

The only intrinsic parameters that it may be possible to obtain are the photosite dimensions ρ_w and ρ_h from the sensor manufacturer's data sheet.

common experience that the intrinsic parameters change if a lense is detached and reattached, or adjusted for focus or aperture.

AncoraSIR.com

SUSTech Southern University of Science and Technology

Camera Calibration

Some are done before shipping, some are not, and some are provided with a software to do so

- The process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system

Disregard overall scaling, set to 1

$$\tilde{p} = C\tilde{P} \qquad \tilde{p} = (u, v, 1)$$

$$C_{11}X + C_{12}Y + C_{13}Z + C_{14} - C_{31}uX - C_{32}uY - C_{33}uZ - \boxed{C_{34}}u = 0$$

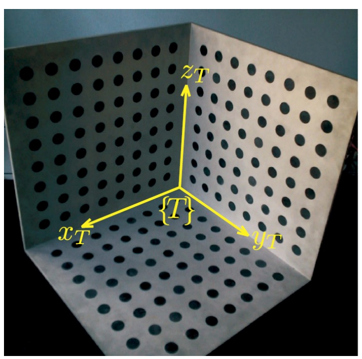
$$C_{21}X + C_{22}Y + C_{23}Z + C_{24} - C_{31}vX - C_{32}vY - C_{33}vZ - C_{34}v = 0$$

$$u = \frac{u'}{w'}, \; v = \frac{v'}{w'}$$

(u_i, v_i)

Increasing sampling for a solution

11 unknows to be solved



$$\begin{pmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & & \vdots & & & & & \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -u_NX_N & -u_NY_N & -u_NZ_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -v_NX_N & -v_NY_N & -v_NZ_N \end{pmatrix} \boxed{\begin{pmatrix} C_{11} \\ C_{12} \\ \vdots \\ C_{33} \end{pmatrix}} = \begin{pmatrix} u_1 \\ v_1 \\ \vdots \\ u_N \\ v_N \end{pmatrix}$$

$N \geq 6$ for *a* solution, but usually more are used to solve using *least square*

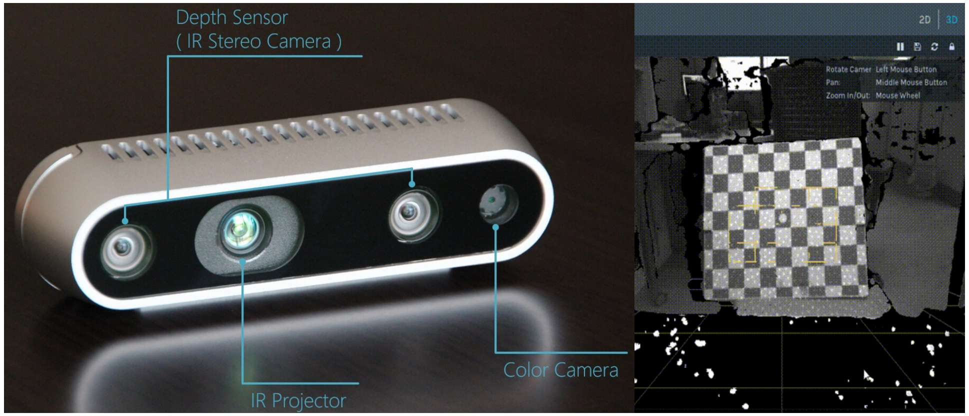
What if the points are coplanar?

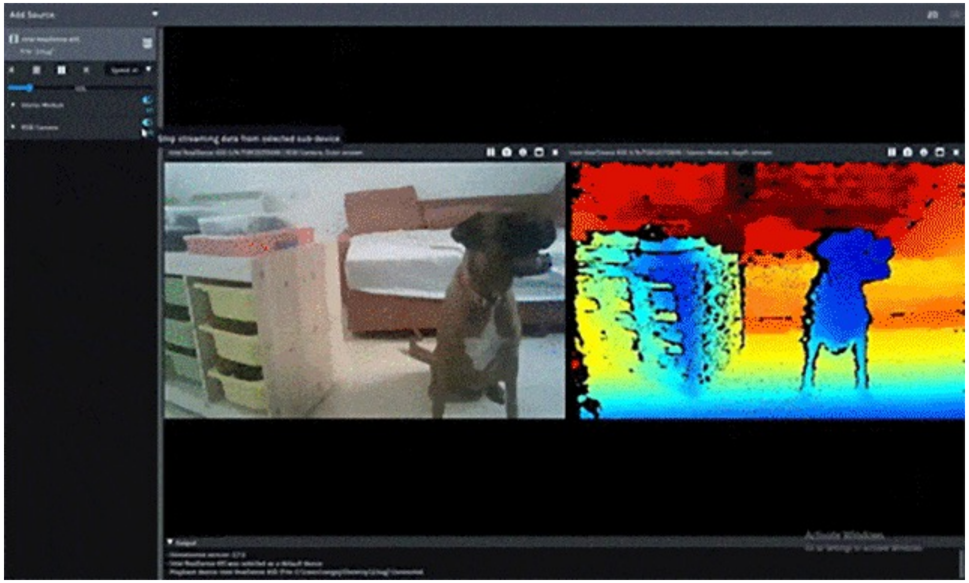
AncoraSIR.com

About Intel Realsense D435

Entry level stereo depth sensor with abundant resources at a low cost

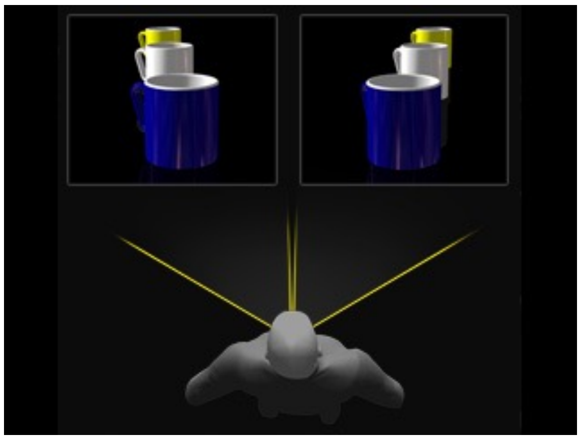
	Intel RealSense Depth Camera D435
Environment	Indoor and outdoor
Depth Technology	Active IR stereo
Image Sensor Technology	Global shutter: 3 um x 3 um pixel size
Main Intel® RealSense™ Products	Intel® RealSense™ vision processor D4 Intel® RealSense™ module D430
Depth Field of View (FOV)—(Horizontal × Vertical) for HD 16:9	85.2° x 58° (+/- 3°)
Depth Stream Output Resolution	Up to 1280 x 720
Depth Stream Output Frame Rate	Up to 90 fps
Minimum Depth Distance (Min-Z)	0.11 m
Maximum Range	Approximately 10 meters Accuracy varies depending on calibration, scene, and lighting conditions
RGB Sensor Resolution & Frame Rate	1920 x 1080 at 30 fps
RGB Sensor FOV (Horizontal × Vertical)	69.4° x 42.5° (+/- 3°)
Camera Dimension (Length x Depth x Height)	90 mm x 25 mm x 25 mm
Connector	USB Type-C*
Mounting Mechanism	One 1/4-20 UNC thread mounting point Two M3 thread mounting points

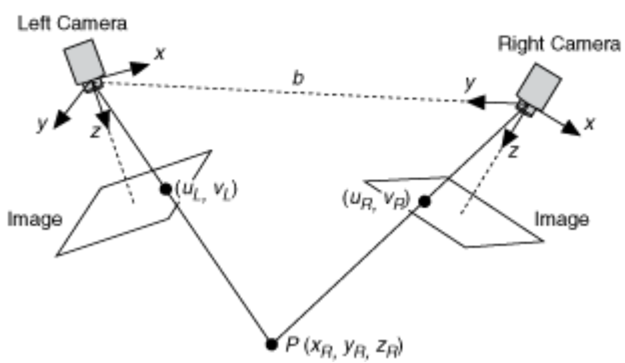


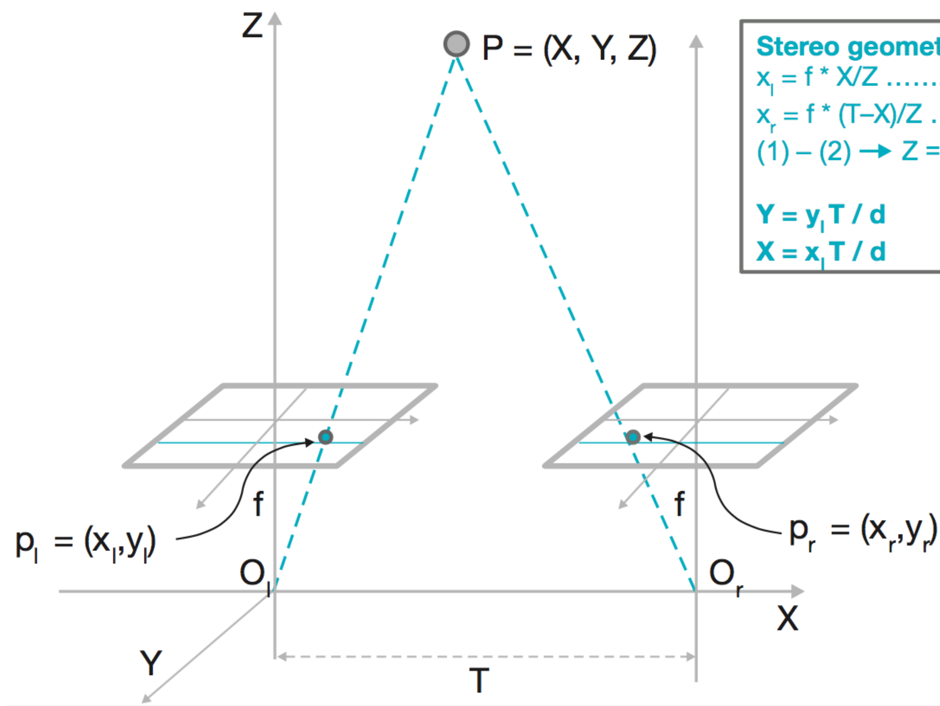


Stereo Vision

Triangulation Principle







Stereo geometry equations

$x_l = f * X/Z$ ……. (1)

$x_r = f * (T-X)/Z$ …(2)

(1) – (2) → $Z = fT/(x_l - x_r)$, if $d = x_l - x_r$

$\mathbf{Y = y_l T / d}$

$\mathbf{X = x_l T / d}$

AncoraSIR.com

Thank you ~

Song Chaoyang
Southern University of Science and Technology