EECE 5550
Mobile Robotics

Lecture 5: Fundamentals of Computer Vision

Derya Aksaray

Assistant Professor

Department of Electrical and Computer Engineering

Northeastern University

Plan of the day

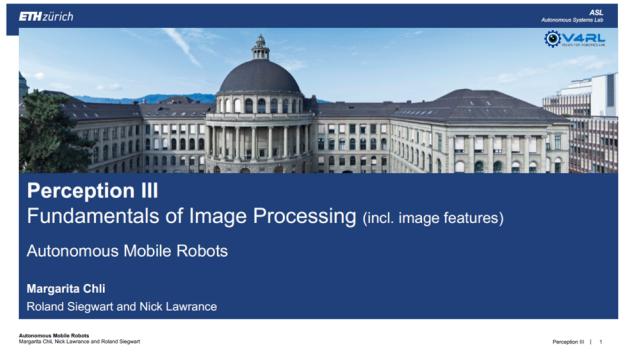
Camera models

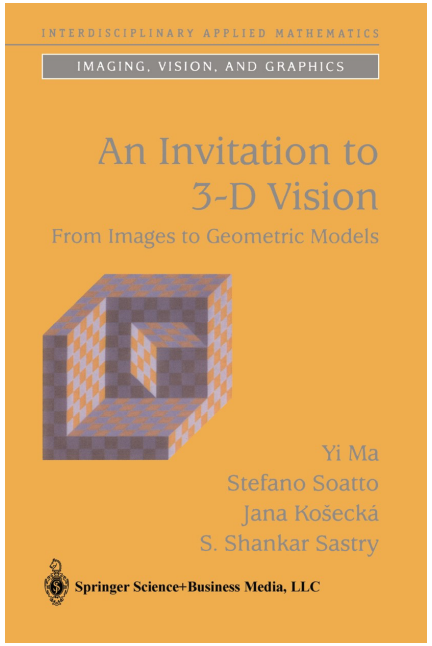
- Image formation
- Pinhole projection model
- Homogeneous coordinates

Feature extraction

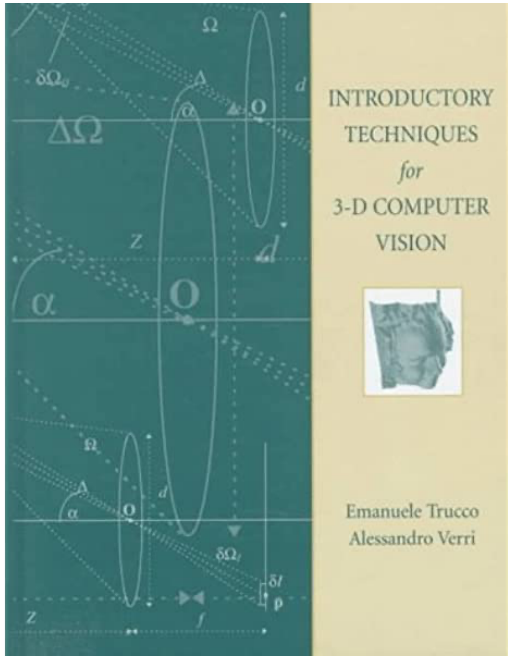
- Noise filtering
- Edge detection
- Corner detection
- April Tags

References





Chapter 3



Chapters 3 & 4

http://graphics.cs.cmu.edu/courses/15-463/lectures/lecture_03.pdf

Representation of images

In the context of computer vision, an image is a 2D brightness array.

In other words, an image is a map I defined on a compact region Ω of a 2D surface and taking values in the positive real numbers.

$$I: \Omega \subset \mathbb{R}^2 \to \mathbb{R}_+; \qquad (x, y) \to I(x, y)$$

- **E.g.,** Ω is a rectangular image sensor in the cameras.

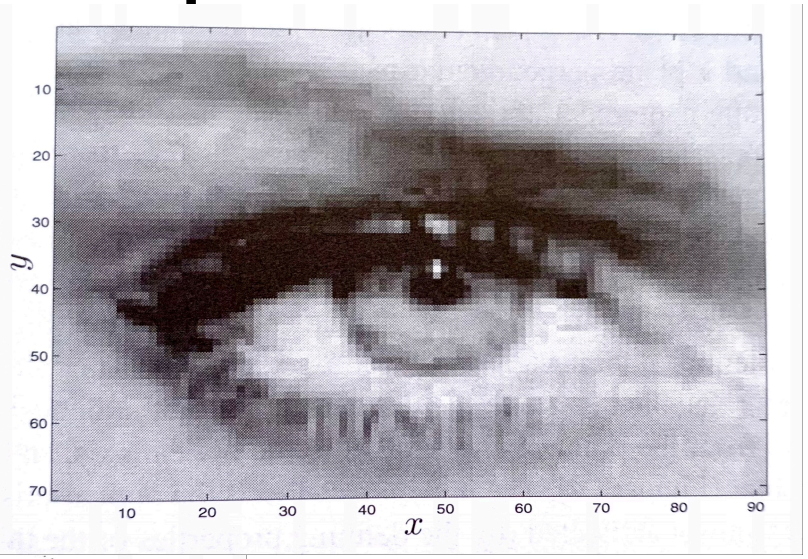
In the case of a digital image, both the domain Ω and the range $\mathbb{R}_+$ are discretized.

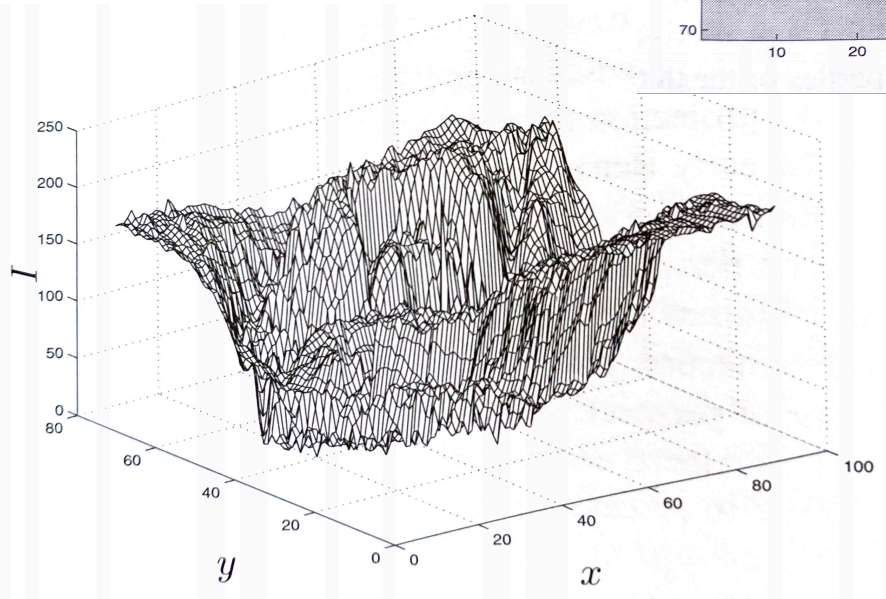
- **E.g.,** $\Omega = [1,640] \times [1,480] \subset \mathbb{Z}^2$ and $\mathbb{R}_+$ is approximated by an interval of integers $[0,255] \subset \mathbb{Z}_+$
- 0 – complete absence of light (black)
- 255 – complete saturation of light (white)

Color	Red number	Green number	Blue number
red	255	0	0
purple	255	0	255
yellow	255	255	0
dark yellow	100	100	0
white	255	255	255
black	0	0	0

Example: Image representations

An image I represented by a picture, 2D surface, and 3D matrix of integers





- At any point (x, y), the value of $I(x, y)$ is called the **image intensity** or **brightness** or **irradiance**.
- Its unit is power per unit area (W/m²).

188	186	188	187	168	130	101	99	110	113	112	107	117	140	153	153	156	158	156	153
189	189	188	181	163	135	109	104	113	113	110	109	117	134	147	152	156	163	160	156
190	190	188	176	159	139	115	106	114	123	114	111	119	130	141	154	165	160	156	151
190	188	188	175	158	139	114	103	113	126	112	113	127	133	137	151	165	156	152	145
191	185	189	177	158	138	110	99	112	119	107	115	137	140	135	144	157	163	158	150
193	183	178	164	148	134	118	112	119	117	118	106	122	139	140	152	154	160	155	147
185	181	178	165	149	135	121	116	124	120	122	109	123	139	141	154	156	159	154	147
175	176	176	163	145	131	120	118	125	123	125	112	124	139	142	155	158	158	155	148
170	170	172	159	137	123	116	114	119	122	126	113	123	137	141	156	158	159	157	150
171	171	173	157	131	119	116	113	114	118	125	113	122	135	140	155	156	160	160	152
174	175	176	156	128	120	121	118	113	112	123	114	122	135	141	155	155	158	159	152
176	174	174	151	123	119	126	121	112	108	122	115	123	137	143	156	155	152	155	150
175	169	168	144	117	117	127	122	109	106	122	116	125	139	145	158	156	147	152	148
179	179	180	155	127	121	118	109	107	113	125	133	130	129	139	153	161	148	155	157
176	183	181	153	122	115	113	106	105	109	123	132	131	131	140	151	157	149	156	159
180	181	177	147	115	110	111	107	107	105	120	132	133	133	141	150	154	148	155	157
181	174	170	141	113	111	115	112	113	105	119	130	132	134	144	153	156	148	152	151
180	172	168	140	114	114	118	113	112	107	119	128	130	134	146	157	162	153	153	148
186	176	171	142	114	114	116	110	108	104	116	125	128	134	148	161	165	159	157	149
185	178	171	138	109	110	114	110	109	97	110	121	127	136	150	160	163	158	156	150

*Sub-sampled

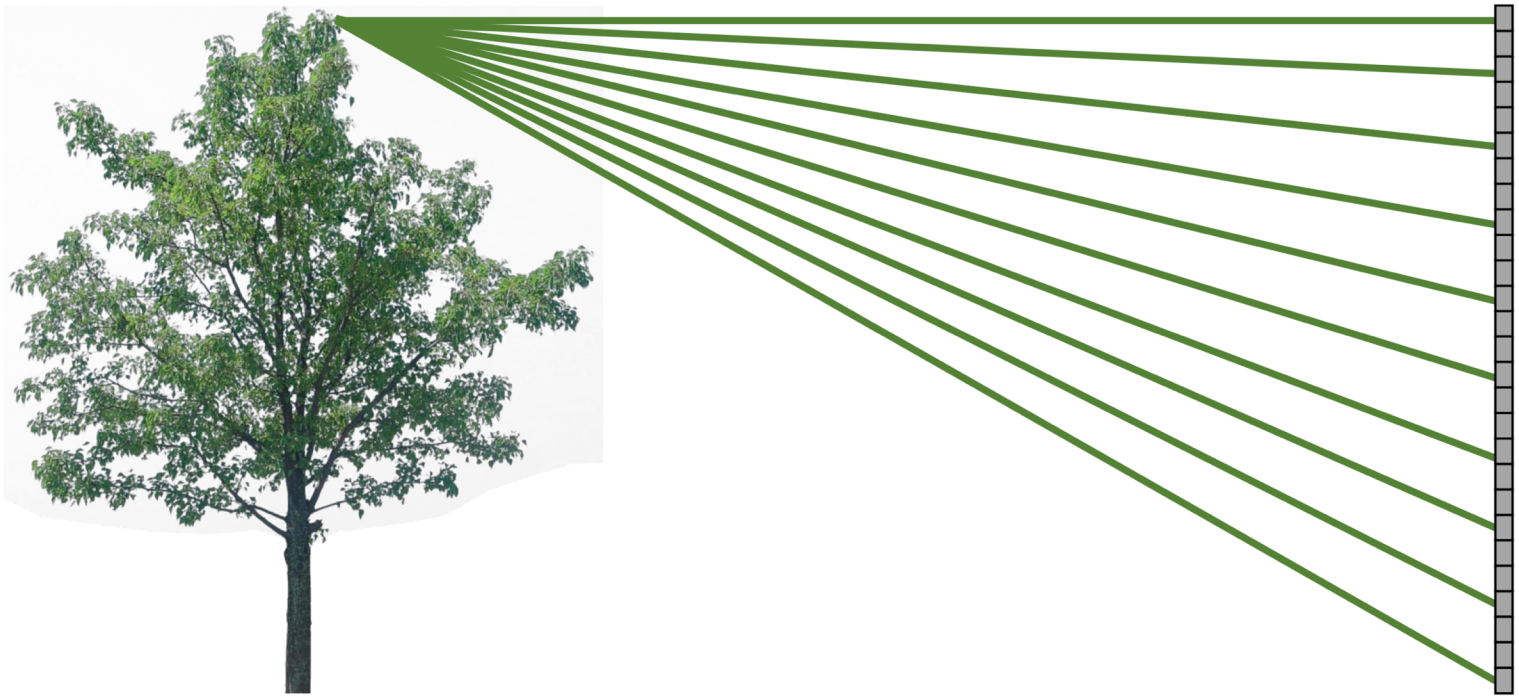
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)

Bare-sensor imaging

real-world object

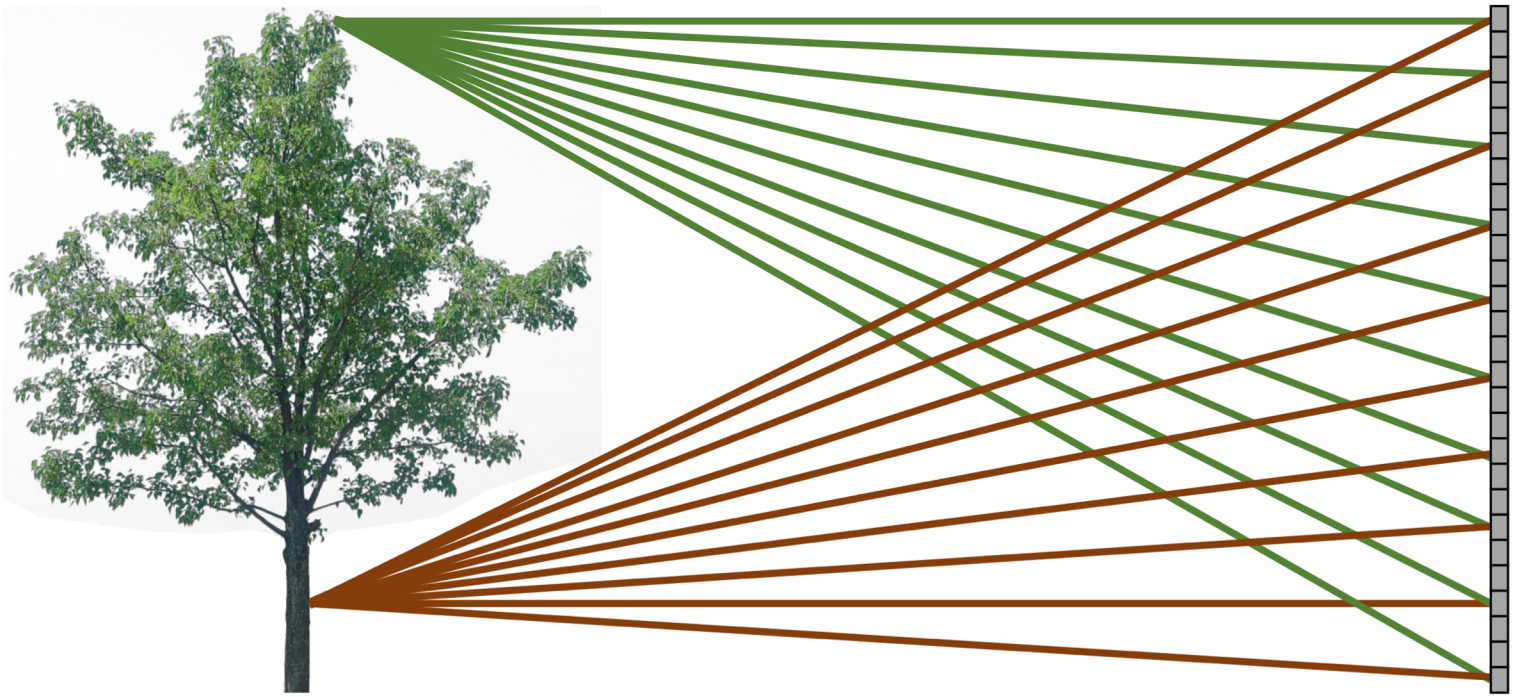


digital sensor (CCD or CMOS)

Bare-sensor imaging

All scene points contribute to all sensor pixels.

real-world object



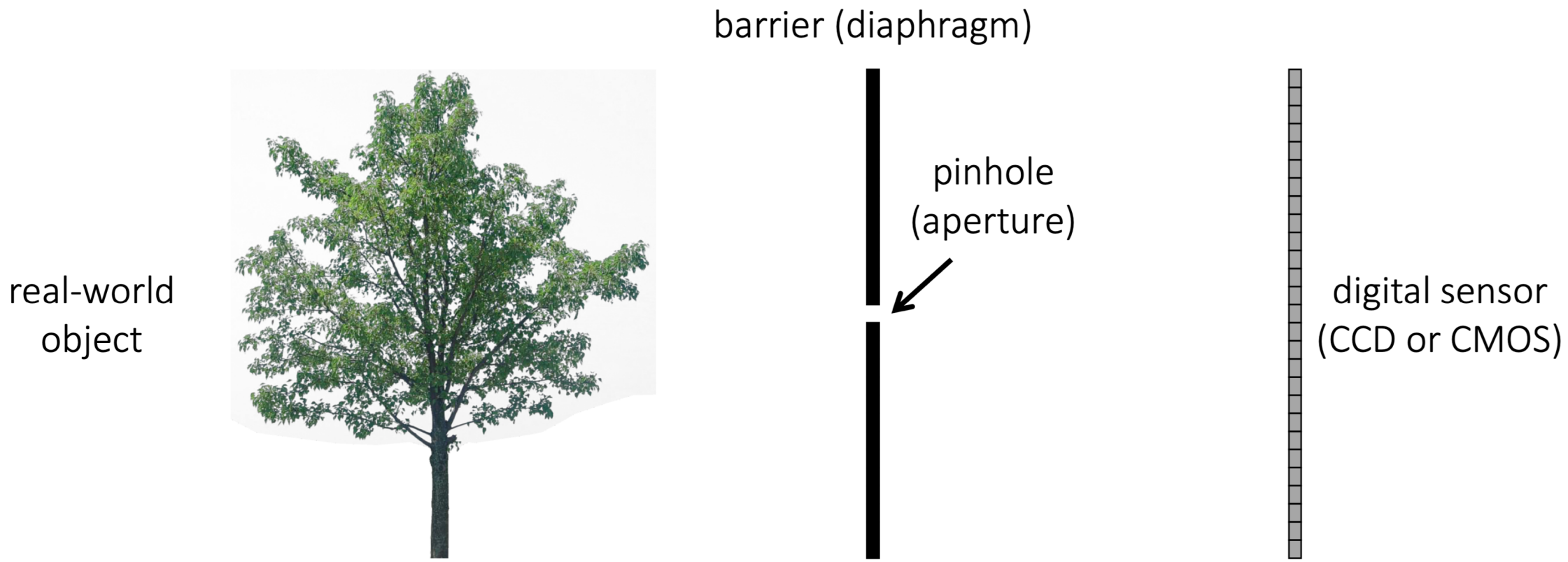
digital sensor (CCD or CMOS)

What is the image on the sensor?

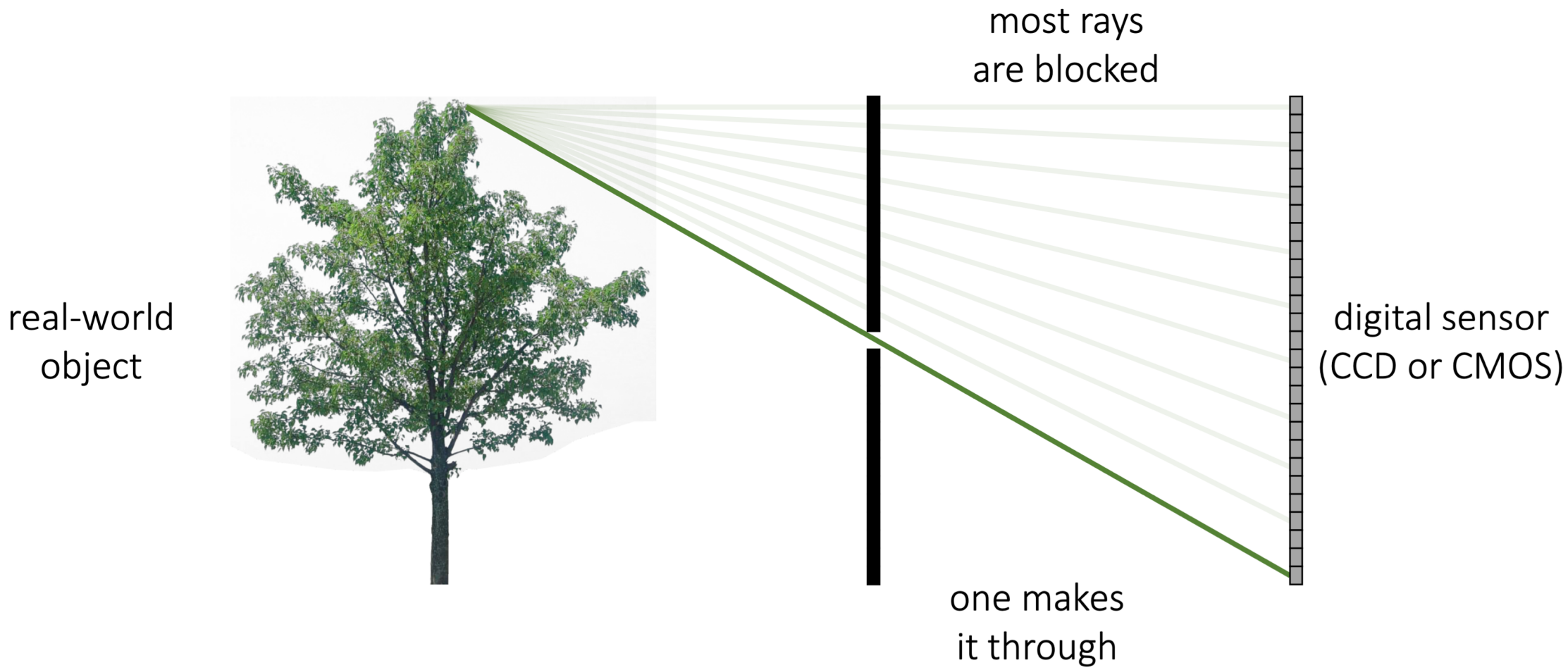
Bare-sensor imaging

The image on the sensor.

Pin-hole imaging

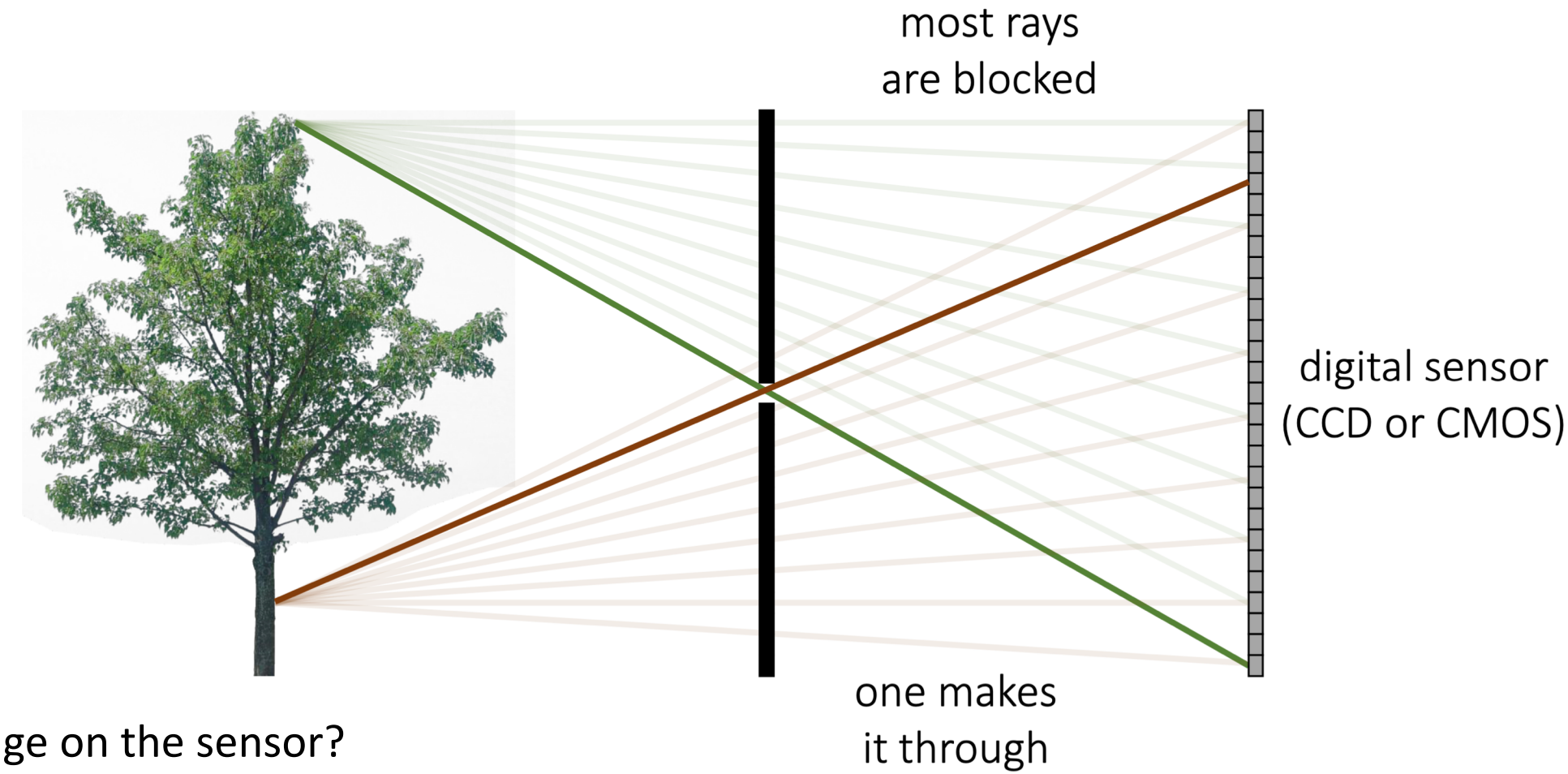


Pin-hole imaging



Pin-hole imaging

Each scene point contributes to only one sensor pixel.

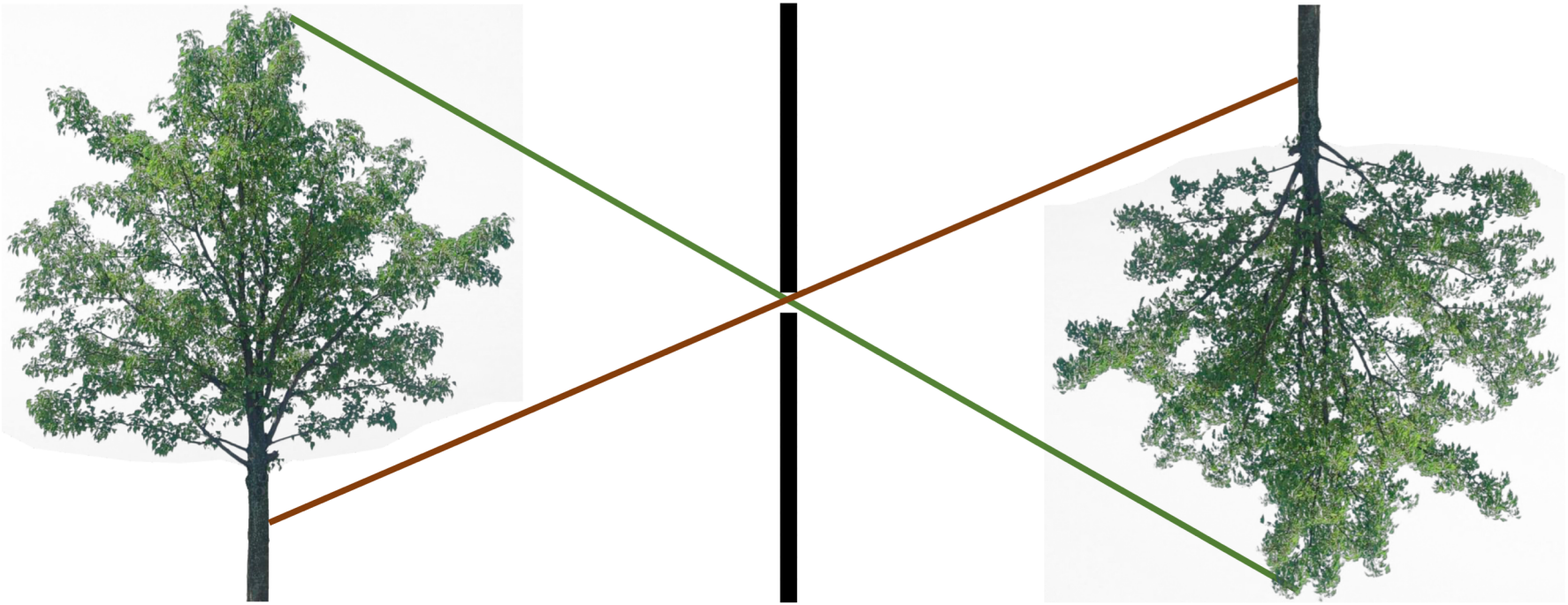


What is the image on the sensor?

Pin-hole imaging

The image on the sensor.

real-world object



copy of real-world object (inverted and scaled)

The camera obscura

Earliest existent written record - Chinese philosopher Mozi (470-390 BC)

Aristotle (384-322 BC), Euclid (323-283 BC)

Discussed the phenomena

Arab mathematician and astronomer Alhazen (965-1039)

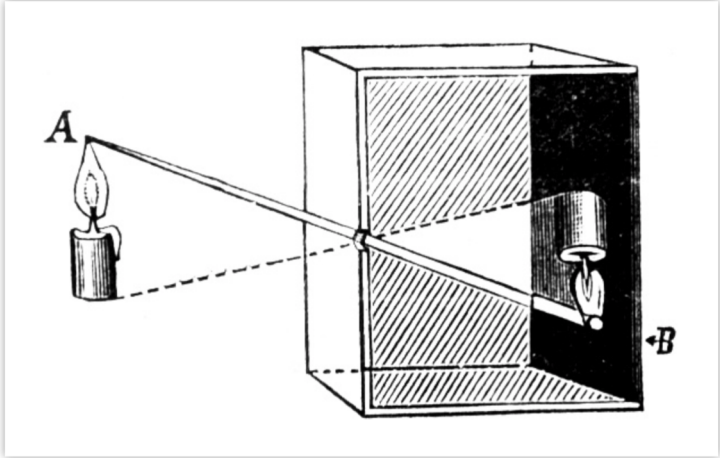
Described the principles of optics

Leonardo da Vinci (1452–1519)

Studied perspective and created real paintings

Image is inverted.

Depth of the room (box) is the effective focal length.



The pinhole camera model

real-world object

barrier (diaphragm)

image plane

pinhole (aperture)

digital sensor (CCD or CMOS)

camera center (center of projection)

The pinhole camera model

What happens as we change the focal length?

object projection is half the size

real-world object

focal length 0.5 f

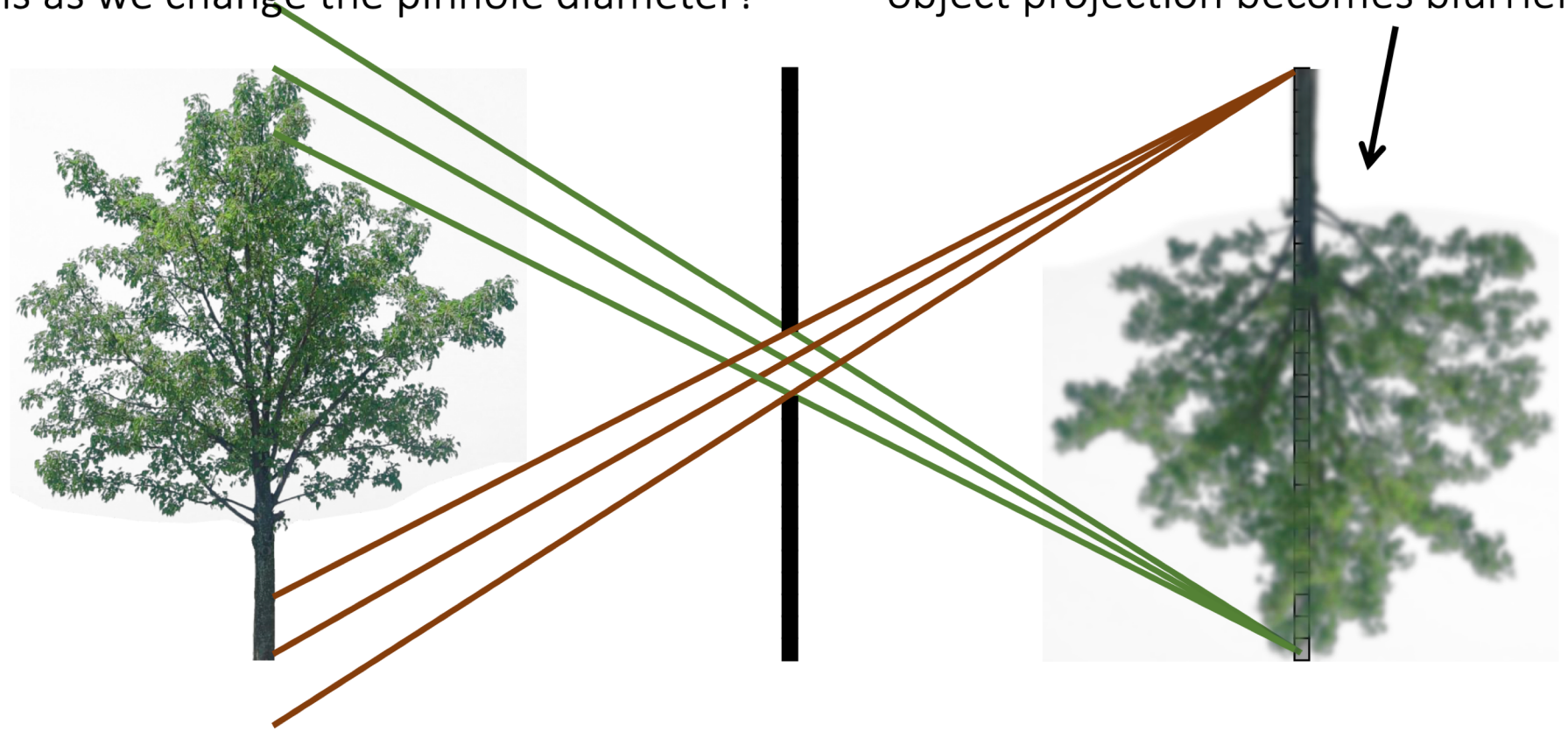
The pinhole camera model

Reducing the size of aperture has also a limit due to the diffraction effects (limiting resolution) and the reduced amount of light.

What happens as we change the pinhole diameter?

object projection becomes blurrier

real-world object

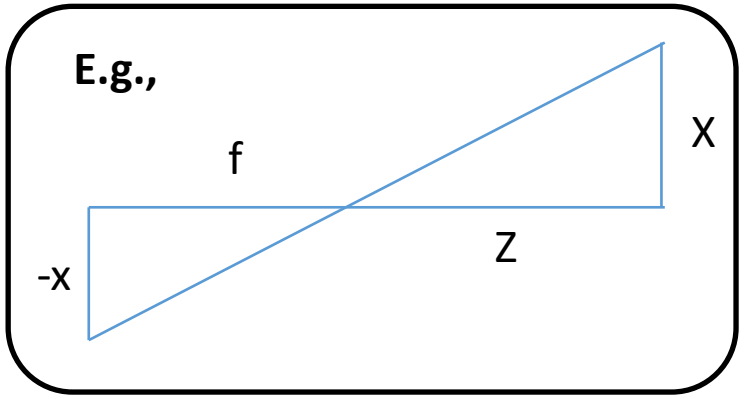


Accidental pinhole camera

https://people.csail.mit.edu/torralba/research/accidentalcameras/videos/shadows_cvpr2012.pdf

Pinhole camera model

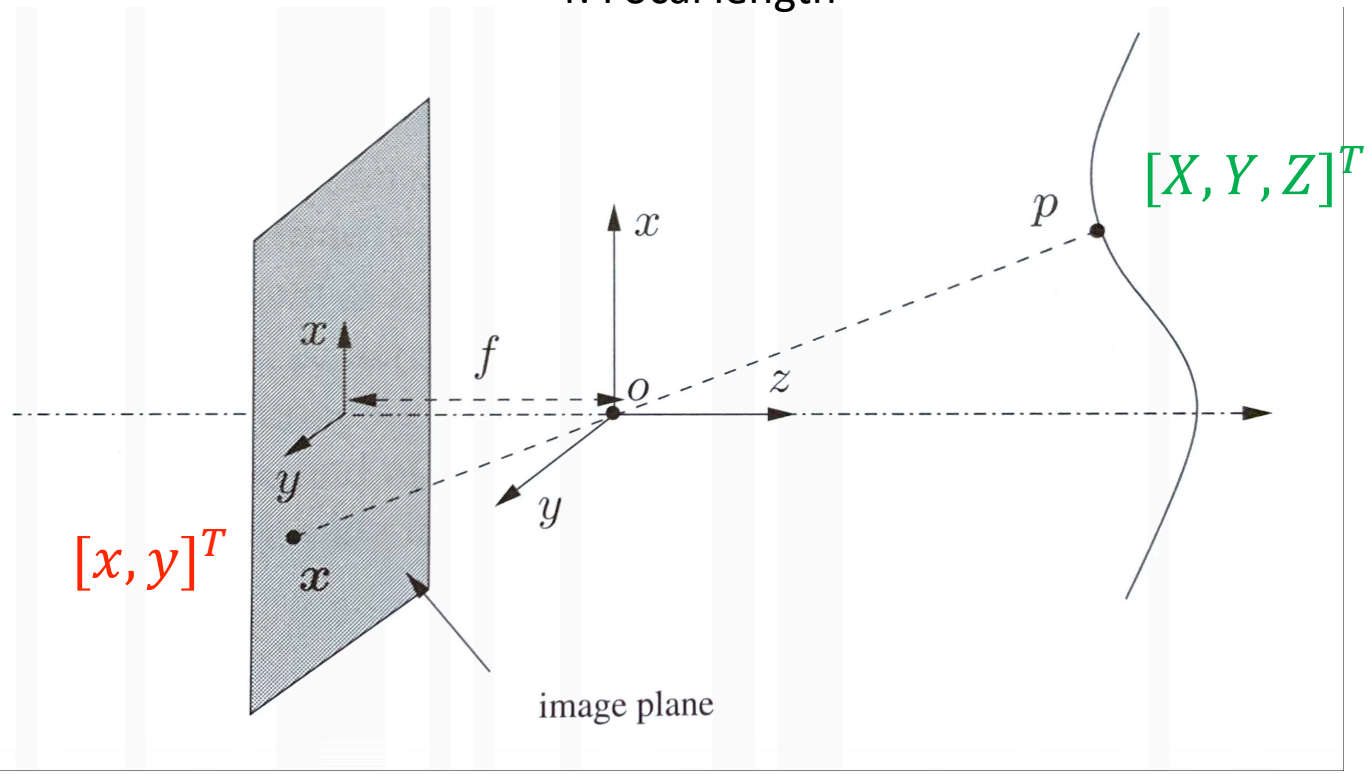
From similar triangles,



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

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

f: Focal length



Perspective projection

Perspective projection

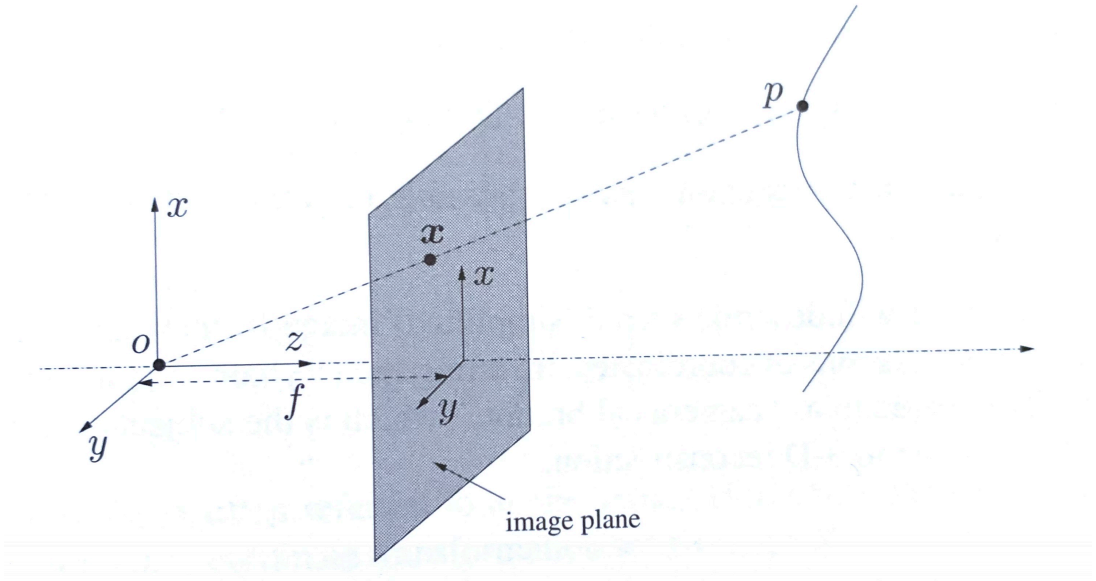
$$x = -f\frac{X}{Z} \qquad y = -f\frac{Y}{Z}$$ ➡ $(x, y) \rightarrow (-x, -y)$ ➡ Flipping the image

- This will be equivalent to placing the image plane $\{z = -f\}$ in front of the optical center $\{z = +f\}$.

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

The point $\boldsymbol{x}$ on the image plane can also be written in homogenous coordinates as:

$$[f\frac{X}{Z}, f\frac{Y}{Z}, 1]^T \in \mathbb{R}^3$$



An ideal perspective camera

Consider a generic point p, with coordinates $\boldsymbol{X_0} = [X_0, Y_0, Z_0]^T \in \mathbb{R}^3$ relative to the world ref. frame.

The same point relative to the camera frame is $\boldsymbol{X} = [X, Y, Z]^T \in \mathbb{R}^3$, which is given by a rigid body transformation $g = (R, T)$ of $\boldsymbol{X_0}$:

$$\boldsymbol{X} = R\boldsymbol{X_0} + T$$

Recall

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \quad \Longleftrightarrow \quad Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Note that

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K_f}\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0};$$

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix};$$

$$\boxed{\lambda\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K_f\Pi_0\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}}$$

Z (depth of p) is usually unknown, so write it as an arbitrary positive scalar λ (usually $\lambda = 1$)

From camera frame to pixel coordinates

The previous derivation is based on the camera frame centered at the optical center with one axis aligned with the optical axis.

In practice, the measurements of the captured image by a digital camera are obtained in terms of pixels (i, j) with the origin of the image coordinate frame in the upper-left corner of the image.

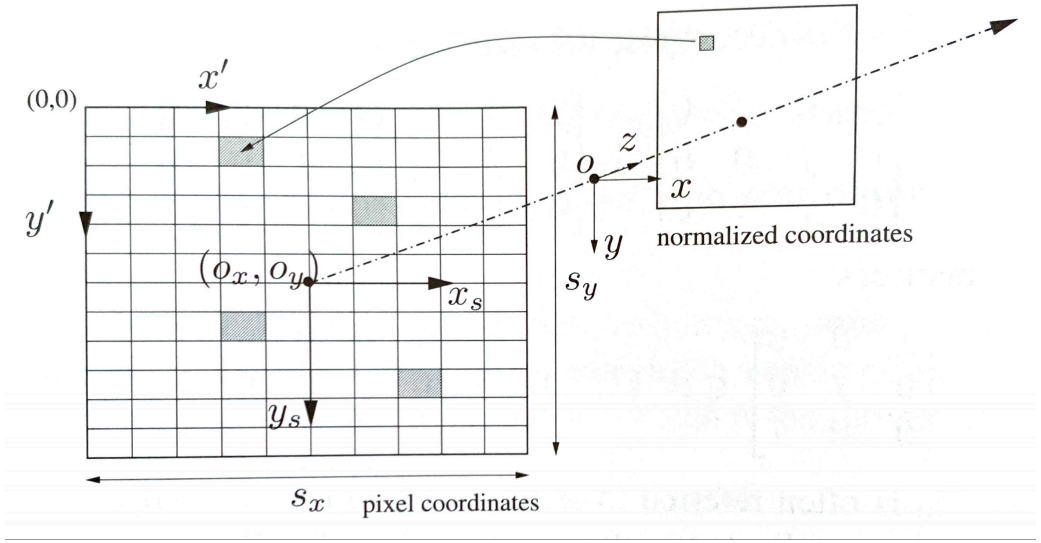
We need to specify the relationship between the camera frame and pixel coordinates.

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}}_{\text{Scaling matrix depending on the size of the pixel}} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$x' = x_s + o_x$$

$$y' = y_s + o_y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



From camera frame to pixel coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

In general, the scaling matrix can be considered as $\begin{bmatrix} s_x & s_\theta \\ 0 & s_y \end{bmatrix}$ where s_θ is the skew factor.

Recall:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Intrinsic parameter matrix
Calibration matrix

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

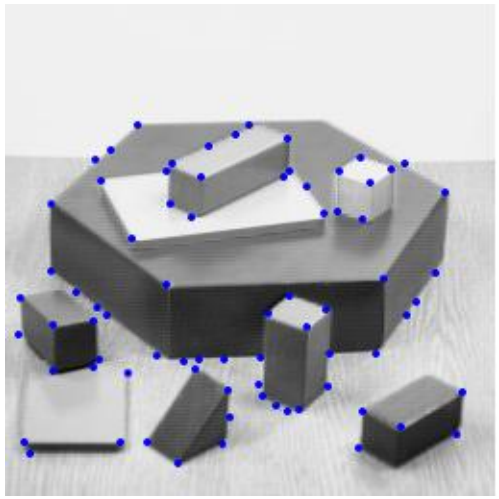
$$= \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

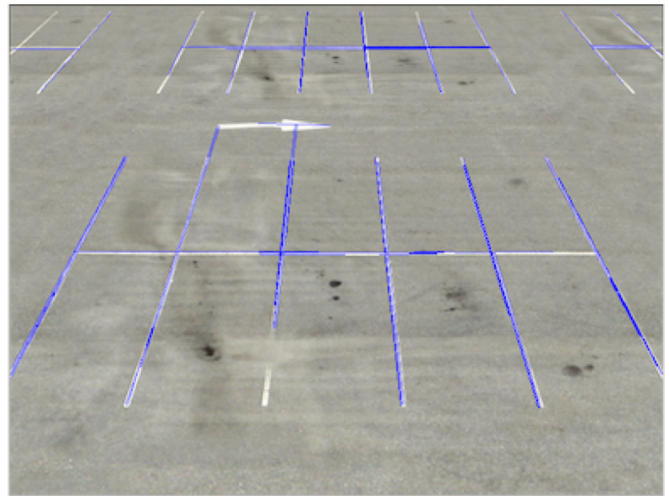
Image Features

Image feature is a part of the image with some special properties.

used to capture “interesting” elements of the scene.

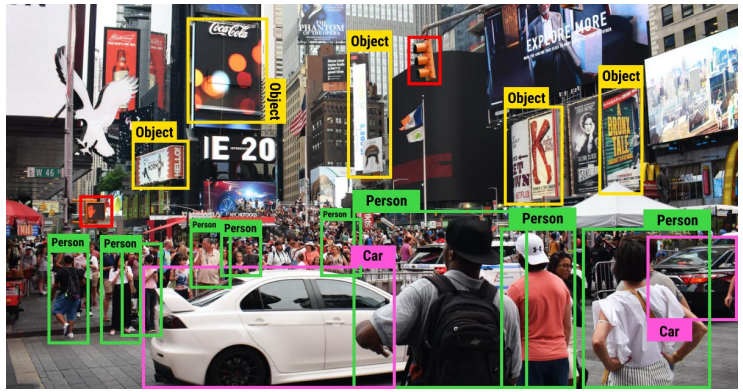
Examples: edges, corners, lines…





Uses of Image Features

Images features are used in many applications such as object recognition, visual tracking, pose estimation, 3D reconstruction, vision-based control/navigation, …



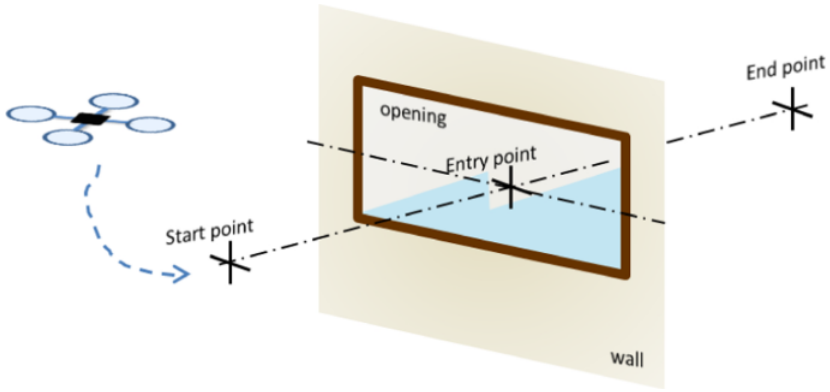


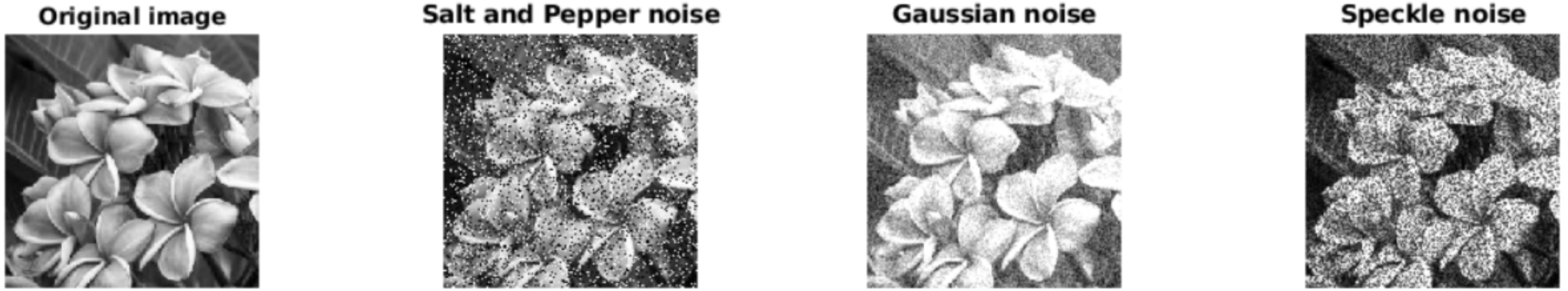
Image Noise

Image is often corrupted by noise during acquisition/transmission.

- sources: low light, quantization, transmission errors, …
- can be additive or multiplicative

Additive noise models: $I_n(x, y) = I(x, y) + N(x, y)$

- $N(x, y)$ is a random signal



Noise filtering

- **Goal:** Recovering $I(x, y)$ from a given noisy image $I_n(x, y)$.
- An estimated image $\hat{I}(x, y)$ is obtained by using a linear (e.g., mean, Gaussian) or non-linear (e.g., median) filter on $I_n(x, y)$.
- Value of $\hat{I}(x, y)$ is obtained as a local function on $I_n(x, y)$

Reference Image

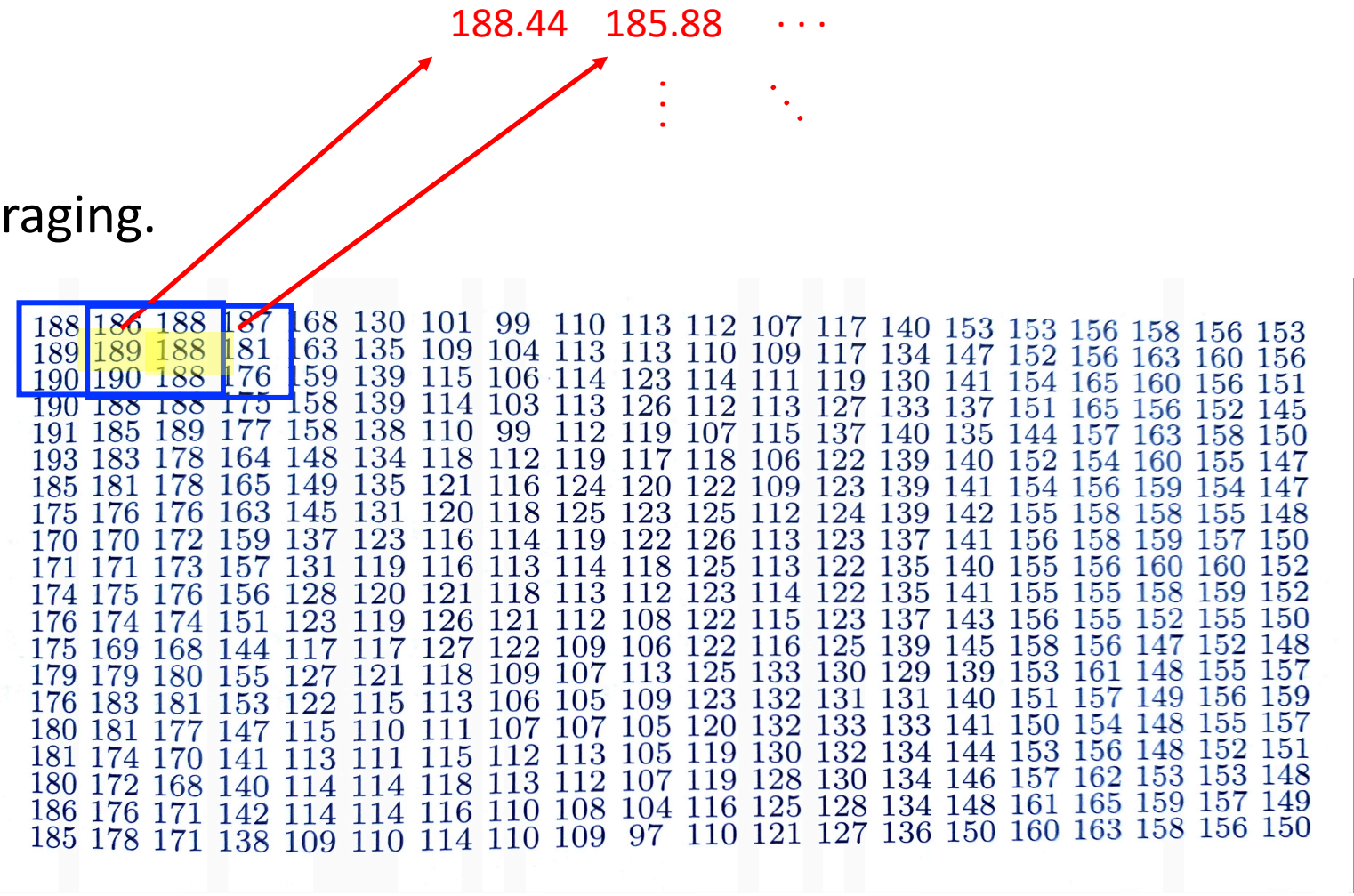
Noisy Image

Denoised Image

Noise filtering

Linear filters: local weighted averaging.

$$\text{Mean: } \begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$$

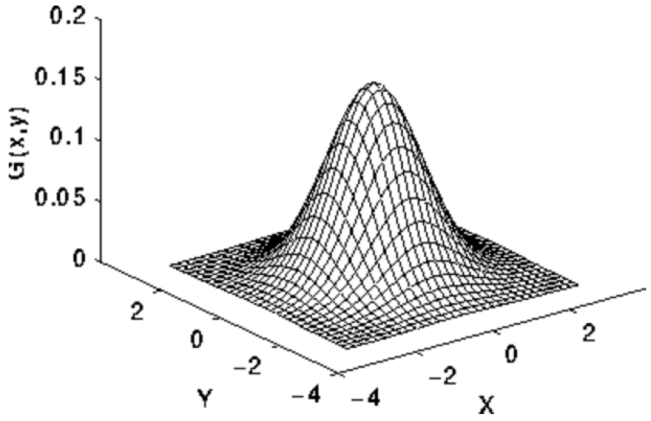


Noise filtering

Linear filters: local weighted averaging.

Gaussian smoothing

188	186	188	187	168	130	101	99	110	113	112	107	117	140	153	153	156	158	156	153
189	189	188	181	163	135	109	104	113	113	110	109	117	134	147	152	156	163	160	156
190	190	188	176	159	139	115	106	114	123	114	111	119	130	141	154	165	160	156	151
190	188	188	175	158	139	114	103	113	126	112	113	127	133	137	151	165	156	152	145
191	185	189	177	158	138	110	99	112	119	107	115	137	140	135	144	157	163	158	150
193	183	178	164	148	134	118	112	119	117	118	106	122	139	140	152	154	160	155	147
185	181	178	165	149	135	121	116	124	120	122	109	123	139	141	154	156	159	154	147
175	176	176	163	145	131	120	118	125	123	125	112	124	139	142	155	158	158	155	148
170	170	172	159	137	123	116	114	119	122	126	113	123	137	141	156	158	159	157	150
171	171	173	157	131	119	116	113	114	118	125	113	122	135	140	155	156	160	160	152
174	175	176	156	128	120	121	118	113	112	123	114	122	135	141	155	155	158	159	152
176	174	174	151	123	119	126	121	112	108	122	115	123	137	143	156	155	152	155	150
175	169	168	144	117	117	127	122	109	106	122	116	125	139	145	158	156	147	152	148
179	179	180	155	127	121	118	109	107	113	125	133	130	129	139	153	161	148	155	157
176	183	181	153	122	115	113	106	105	109	123	132	131	131	140	151	157	149	156	159
180	181	177	147	115	110	111	107	107	105	120	132	133	133	141	150	154	148	155	157
181	174	170	141	113	111	115	112	113	105	119	130	132	134	144	153	156	148	152	151
180	172	168	140	114	114	118	113	112	107	119	128	130	134	146	157	162	153	153	148
186	176	171	142	114	114	116	110	108	104	116	125	128	134	148	161	165	159	157	149
185	178	171	138	109	110	114	110	109	97	110	121	127	136	150	160	163	158	156	150



$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{1}{273}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline \end{array}$$

Discrete approximation to Gaussian function with σ=1.0

How to pick σ?

- Too small – too little smoothing
- Too large – too much smoothing (blur)

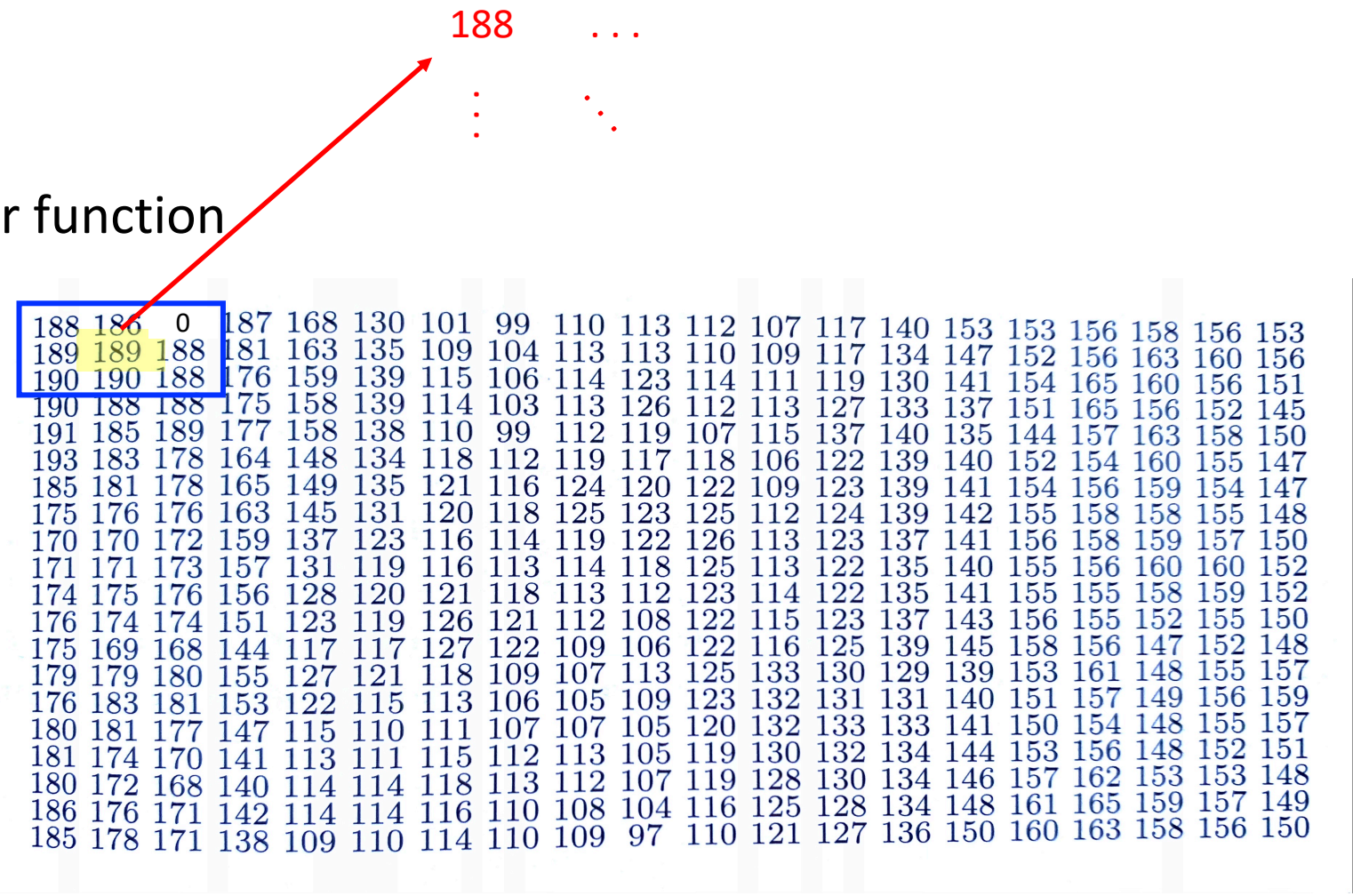
How to pick the mask size?

- Odd number close to 5σ

Noise filtering

Nonlinear filters: local non-linear function

e.g., Median with 3x3 mask



Edge detection

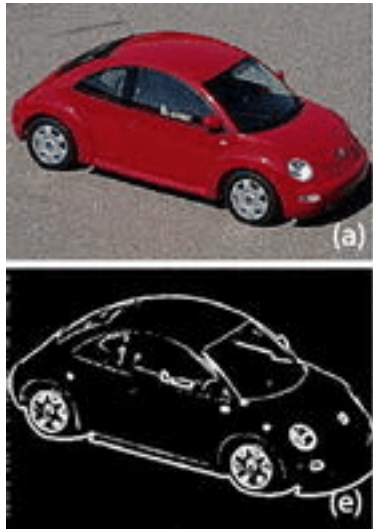
Edges: Pixels where $I(x, y)$ undergoes a sharp variation.

Problem: Given a noisy image $I_n(x, y)$, find the edges in $I(x, y)$.

Algorithms: Canny, Sobel, …

Main steps:

- Noise filtering
- Finding the locations where the magnitude of the gradient is high



Look at the gradient (intensity change)

in x and y directions

12	90	89	86	87	82
10	12	88	85	83	84
9	15	12	84	84	88
12	14	10	82	88	89
11	17	16	12	88	90
10	16	15	17	89	88

Edge detection

Edges: Pixels where $I(x, y)$ undergoes a sharp variation.

Problem: Given a noisy image $I_n(x, y)$, find the edges in $I(x, y)$.

Algorithms: Canny, Sobel, …

Main steps:

- Noise filtering
- Finding the locations where the magnitude of the gradient is high

Sobel

-1	0	+1
-2	0	+2
-1	0	+1

x dir.

+1	+2	+1
0	0	0
-1	-2	-1

y dir.

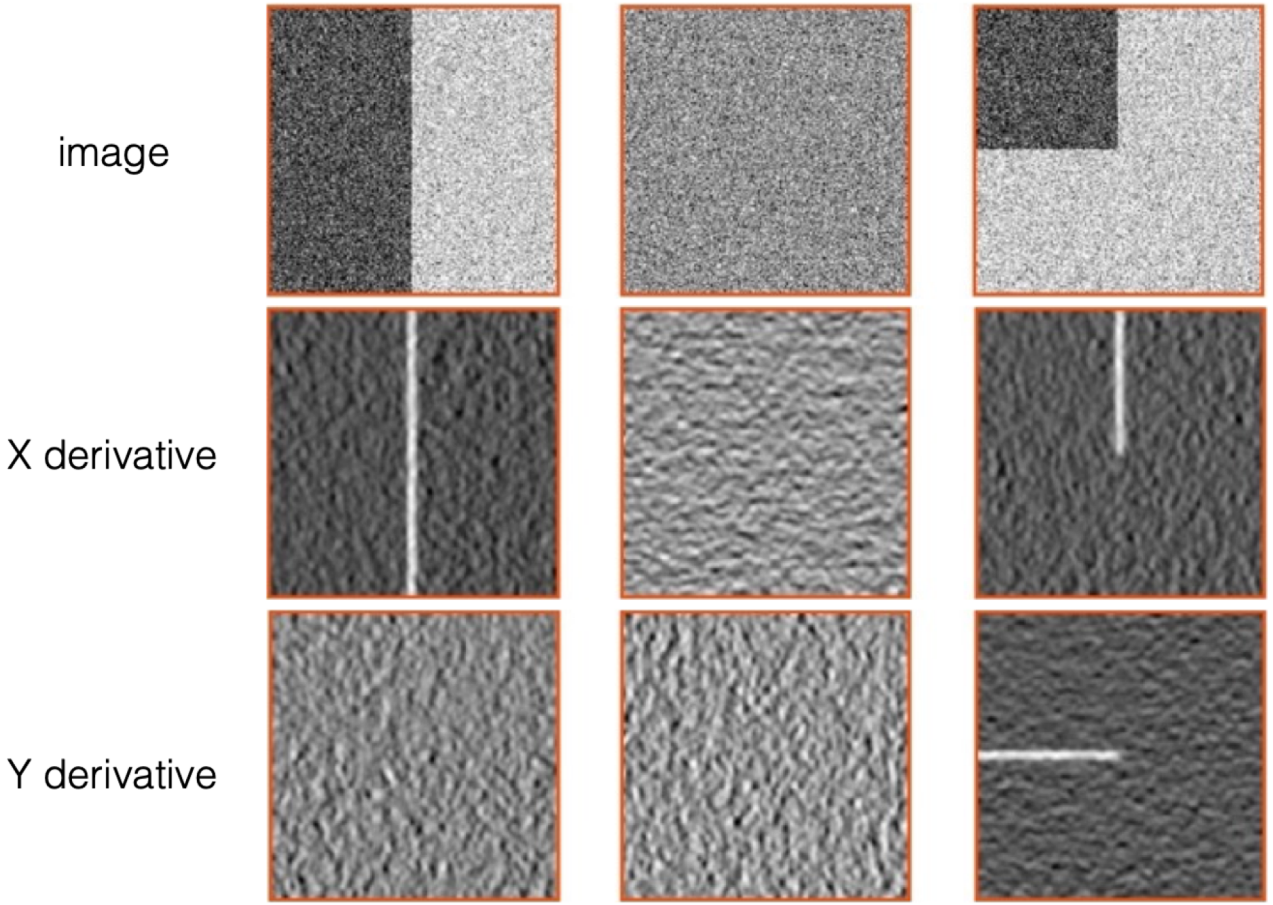
188	186	188	187	168	130	101	99	110	113	112	107	117	140	153	153	156	158	156	153
189	189	188	181	163	135	109	104	113	113	110	109	117	134	147	152	156	163	160	156
190	190	188	176	159	139	115	106	114	123	114	111	119	130	141	154	165	160	156	151
190	188	188	175	158	139	114	103	113	126	112	113	127	133	137	151	165	156	152	145
191	185	189	177	158	138	110	99	112	119	107	115	137	140	135	144	157	163	158	150
193	183	178	164	148	134	118	112	119	117	118	106	122	139	140	152	154	160	155	147
185	181	178	165	149	135	121	116	124	120	122	109	123	139	141	154	156	159	154	147
175	176	176	163	145	131	120	118	125	123	125	112	124	139	142	155	158	158	155	148
170	170	172	159	137	123	116	114	119	122	126	113	123	137	141	156	158	159	157	150
171	171	173	157	131	119	116	113	114	118	125	113	122	135	140	155	156	160	160	152
174	175	176	156	128	120	121	118	113	112	123	114	122	135	141	155	155	158	159	152
176	174	174	151	123	119	126	121	112	108	122	115	123	137	143	156	155	152	155	150
175	169	168	144	117	117	127	122	109	106	122	116	125	139	145	158	156	147	152	148
179	179	180	155	127	121	118	109	107	113	125	133	130	129	139	153	161	148	155	157
176	183	181	153	122	115	113	106	105	109	123	132	131	131	140	151	157	149	156	159
180	181	177	147	115	110	111	107	107	105	120	132	133	133	141	150	154	148	155	157
181	174	170	141	113	111	115	112	113	105	119	130	132	134	144	153	156	148	152	151
180	172	168	140	114	114	118	113	112	107	119	128	130	134	146	157	162	153	153	148
186	176	171	142	114	114	116	110	108	104	116	125	128	134	148	161	165	159	157	149
185	178	171	138	109	110	114	110	109	97	110	121	127	136	150	160	163	158	156	150

Compute $\hat{I}_x(x, y)$ and $\hat{I}_y(x, y)$

$$G(x, y) = \sqrt{\hat{I}_x(x, y)^2 + \hat{I}_y(x, y)^2}$$

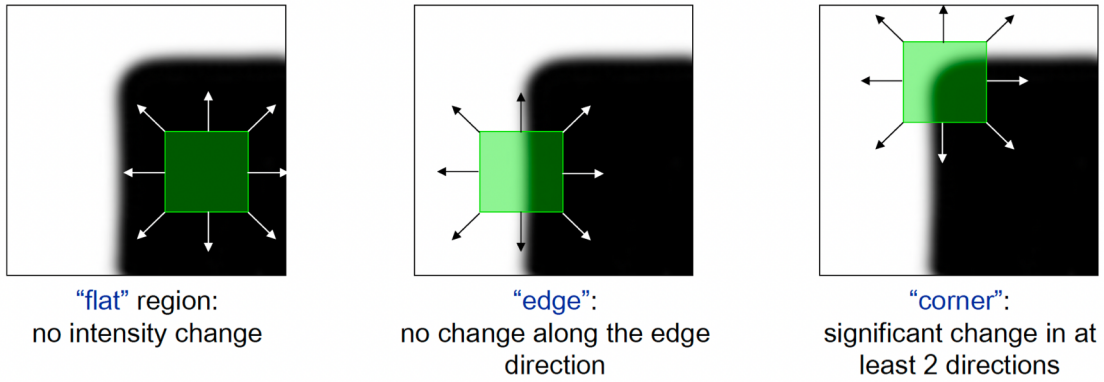
Declare (x, y) is an edge if $G(x, y) > \tau$

Visualization of gradients



Corner detection

Shifting a window in any direction should give a large change of intensity in at least 2 directions



Recall that we have computed $\hat{I}_x(x,y)$ and $\hat{I}_y(x,y)$.
Now, compute

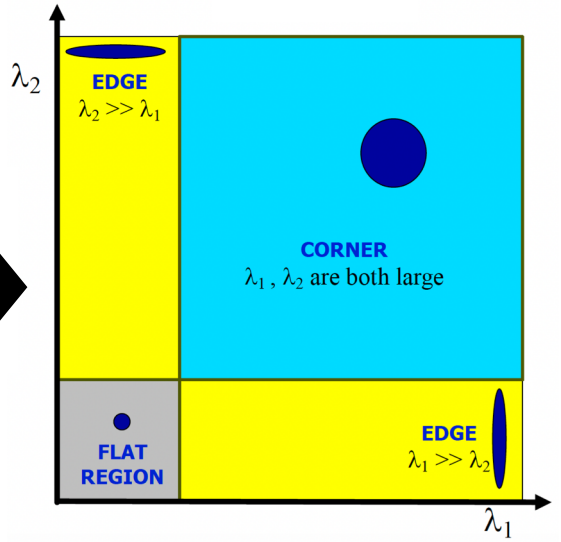
$$C = \begin{bmatrix} \sum \hat{I}_x^2 & \sum \hat{I}_x \hat{I}_y \\ \sum \hat{I}_x \hat{I}_y & \sum \hat{I}_y^2 \end{bmatrix}$$

*sums are taken over a local neighborhood of (x,y)

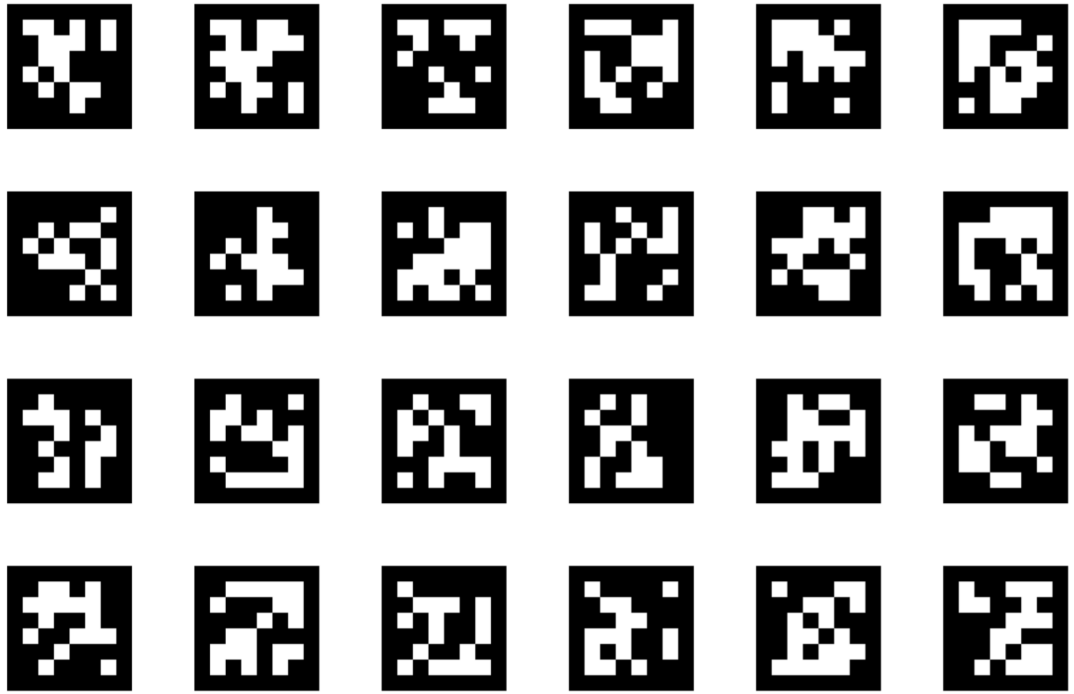
Since C is symmetric,

$$\begin{bmatrix} \sum \hat{I}_x^2 & \sum \hat{I}_x \hat{I}_y \\ \sum \hat{I}_x \hat{I}_y & \sum \hat{I}_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

Look at the values of λ_1 and λ_2



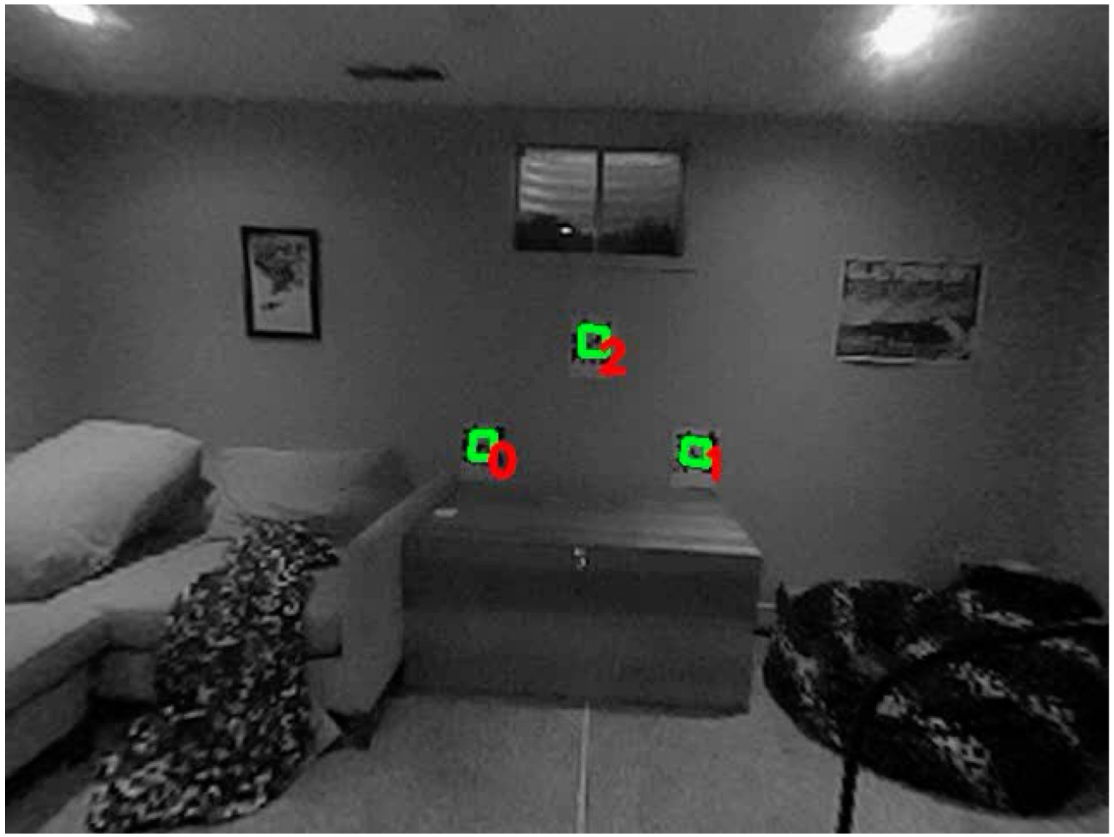
April tags

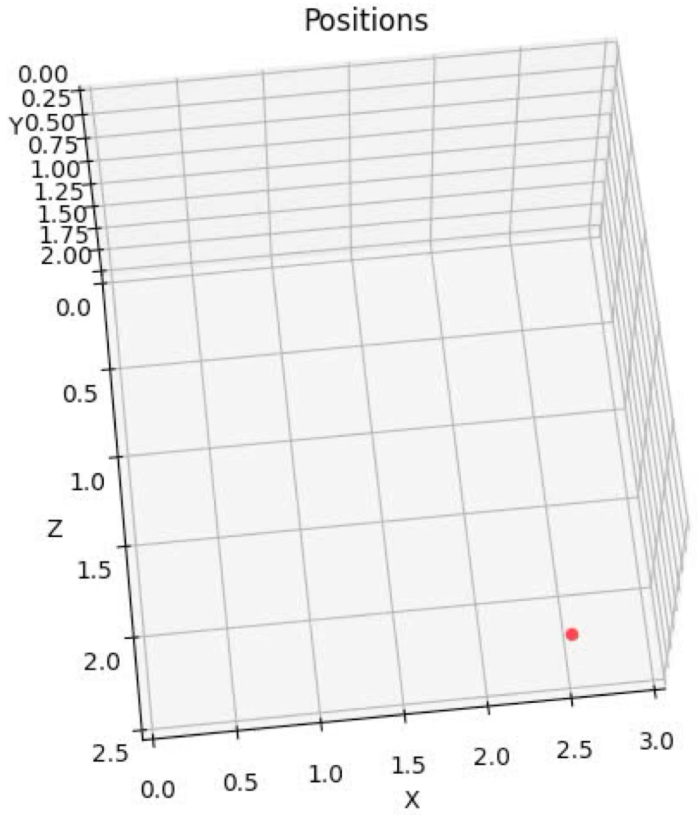


- April tags are visual aids which represent an id using a pattern in black and white.
- Tags having unique ids help in distinguishing objects and recognizing landmarks in the environment.

https://april.eecs.umich.edu/media/pdfs/olson2011tags.pdf

April tags for localization





https://www.youtube.com/watch?v=7iOmVOEvfgw