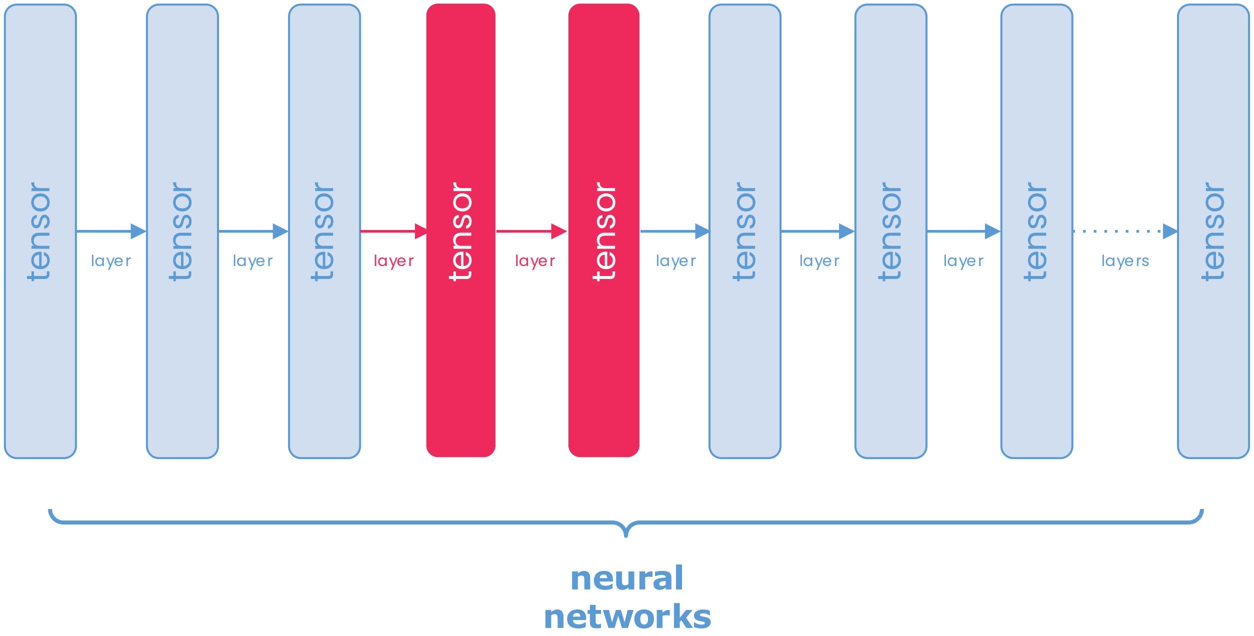
Insper

Computer Vision

Class 9: Corner Detection and Keypoint Matching


tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layers
tensor
neural
networks

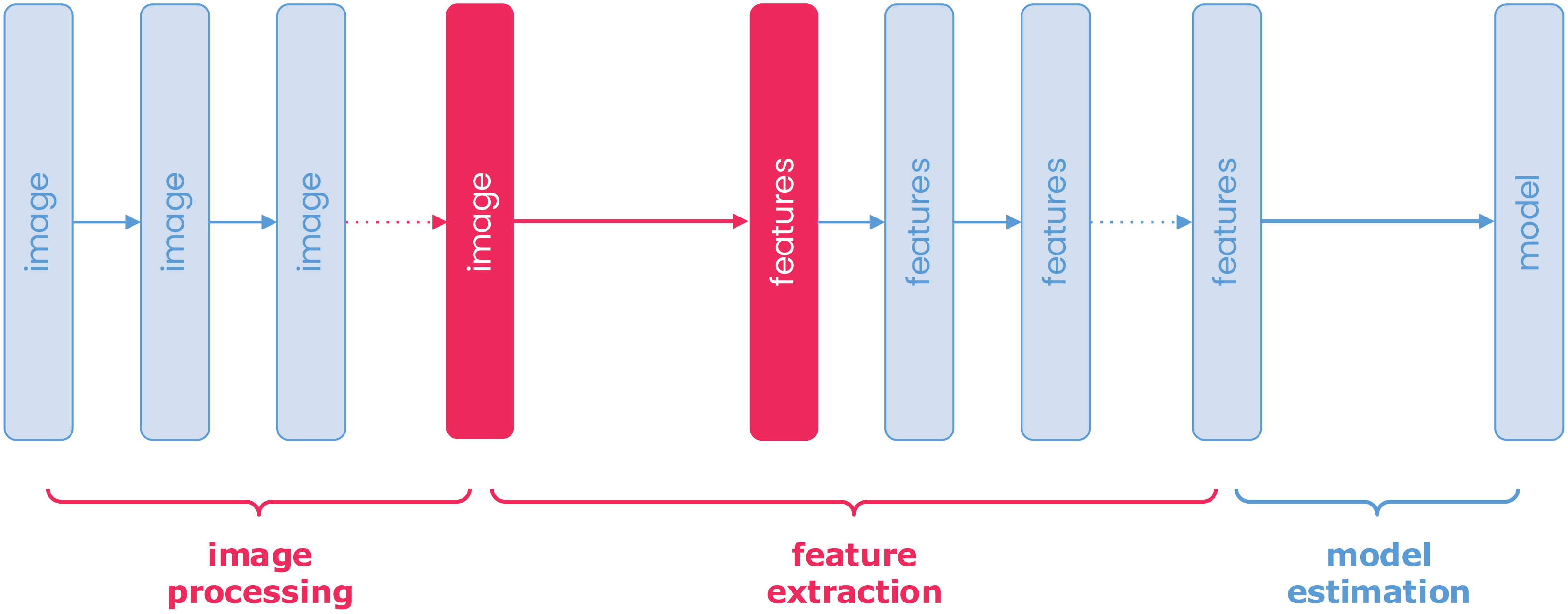
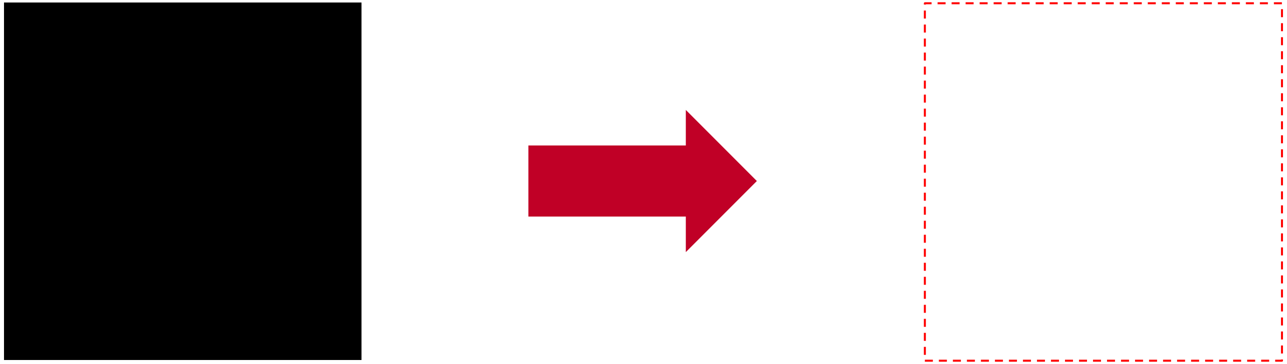
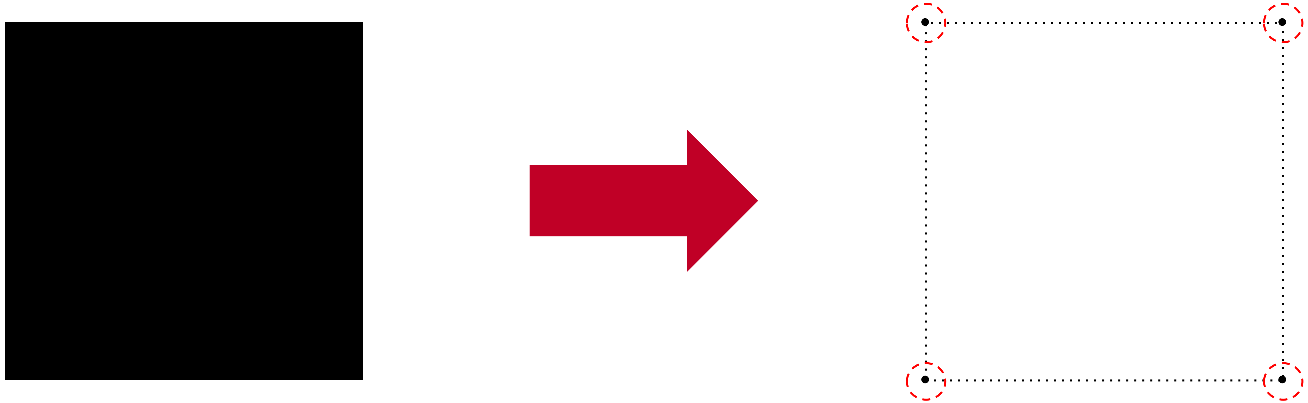


image
image
image
image
features
features
features
features
model
image
processing
feature
extraction
model
estimation

Edge detection:
less pixels while preserving shape information



Corner detection:
even less pixels while preserving shape information



Like edges, corners are...

- ...robust to gray level reduction. *(mostly)*
- ...robust to brightness changes. *(within reason)*
- ...robust to contrast changes. *(within reason)*
- ...robust to occlusion. *(locally)*
- ...robust to translation.
- ...robust to rotation.
- ...robust to perspective distortion. *(locally)*
- ...robust to deformation. *(locally)*

And, like edges, corners are not...

- ...robust to noise, but we can mitigate that with image smoothing.
- ...robust to scale, but we can generalize them in the scale space.

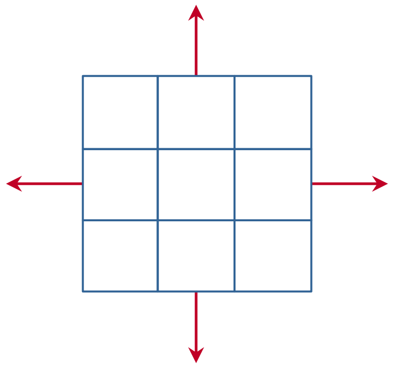
Derivatives do tend to have stronger responses in corners than in edges...

...but the threshold that separates them is dependent on the context.

Let's try a non-convolutional approach.

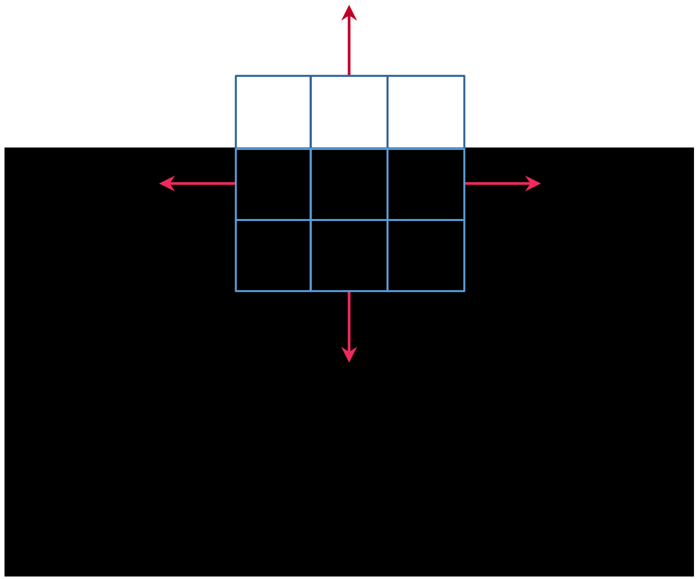
Moravec (1980):

the more a window changes after small displacements,
the more the center of this window is likely to be a corner



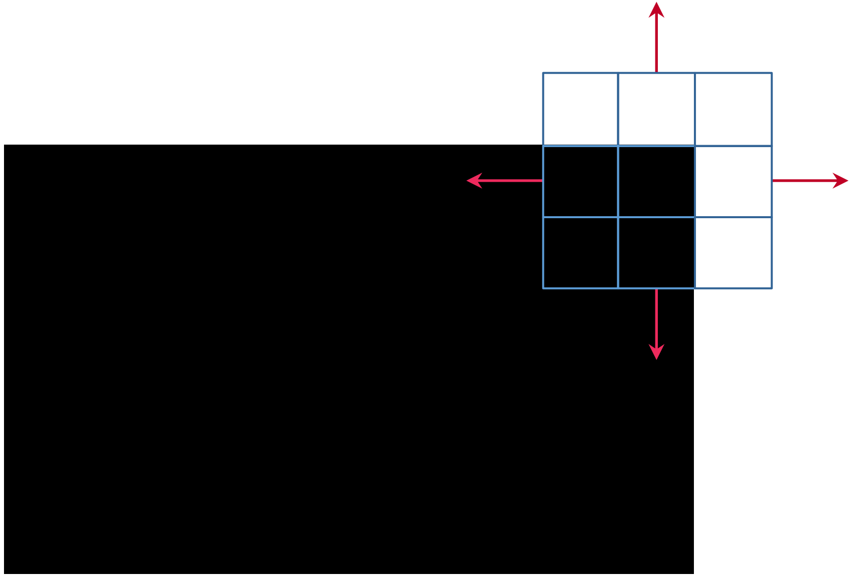
Moravec (1980):

the more a window changes after small displacements,
the more the center of this window is likely to be a corner



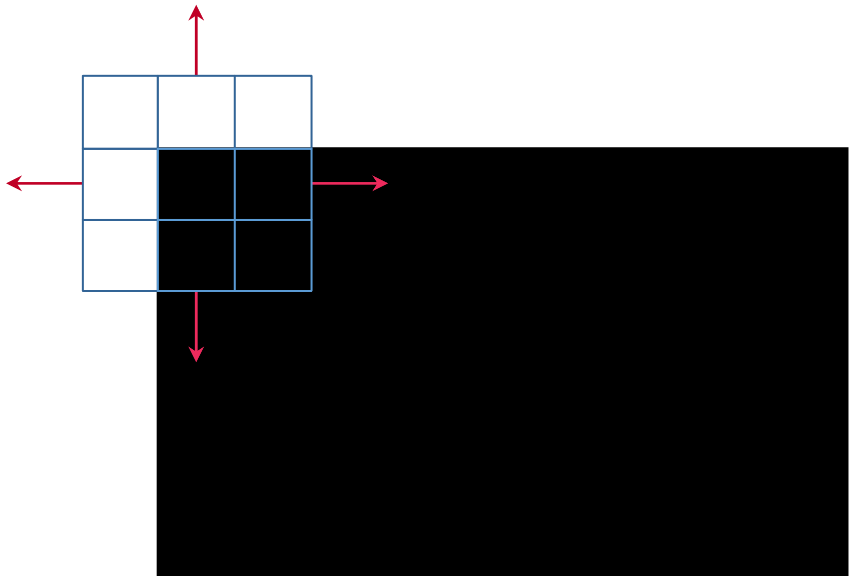
Moravec (1980):

the more a window changes after small displacements,
the more the center of this window is likely to be a corner



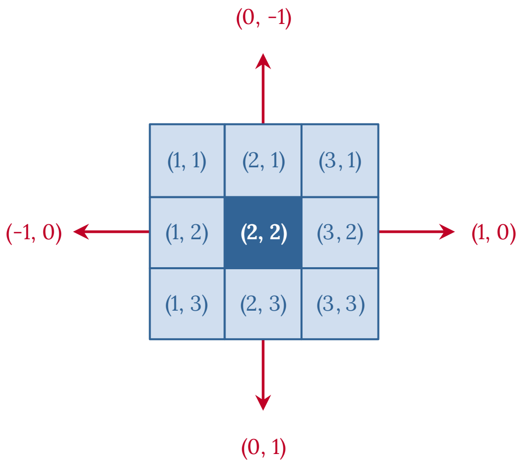
Moravec (1980):

the more a window changes after small displacements,
the more the center of this window is likely to be a corner



Moravec (1980):

the more a window changes after small displacements,
the more the center of this window is likely to be a corner



$\mathcal{D} = \{ (0, -1), (-1, 0), (1, 0), (0, 1) \}$

$\mathcal{W} = \{ (1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3) \}$

Moravec (1980):

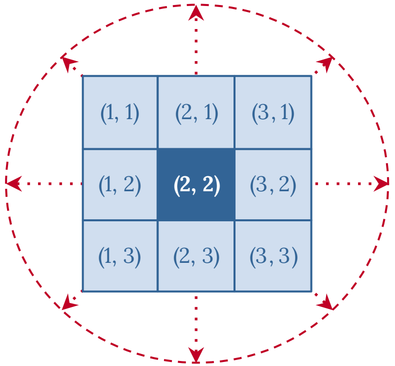
the more a window changes after small displacements,
the more the center of this window is likely to be a corner

$$\sum_{(d_x,d_y)\in\mathcal{D}} \sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2$$

Also has context-dependent threshold.

Harris-Stephens (1988):

instead of considering a discrete set of chosen displacements,
consider a continuous set of all possible small displacements



$\mathcal{D}$ = unit vectors

$\mathcal{W}$ = { (1, 1), (2, 1), (3, 1), (1, 2), **(2, 2)**, (3, 2), (1, 3), (2, 3), (3, 3) }

Harris-Stephens (1988):

and instead of calculating the sum of differences...

$$\cancel{\sum_{(d_x,d_y)\in\mathcal{D}}} \sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2$$

Harris-Stephens (1988):

...calculate which directions maximize the difference

$$\sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2$$

Sobel vertical edge filter:

convolution with a kernel that measures the gradient across the x dimension

P

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

D_x

normalized signed values

$G_x = P * D_x$

Sobel horizontal edge filter:

convolution with a kernel that measures the gradient across the y dimension

P

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

D_y

normalized signed values

$G_y = P * D_y$

Harris-Stephens (1988):

…calculate which directions maximize the difference

$$\underbrace{\sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2 = d^t T d}$$

$$d = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \qquad T = \sum_{(x,y)\in\mathcal{W}} \begin{bmatrix} G_x(x,y)^2 & G_x(x,y)\cdot G_y(x,y) \\ G_x(x,y)\cdot G_y(x,y) & G_y(x,y)^2 \end{bmatrix}$$

Harris-Stephens (1988):

...calculate which directions maximize the difference

$$\sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2$$

$$= d^t T d$$

$$= \underbrace{d \cdot Td}_{\text{dot product}}$$

Harris-Stephens (1988):

...calculate which directions maximize the difference

$$\sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2$$

$$= d^t T d$$

$$= d \cdot T d$$

$$= ||d|| \cdot \underbrace{||Td|| \cdot \cos\theta_d}_{}$$

angle between d and Td

Harris-Stephens (1988):

...calculate which directions maximize the difference

$$\sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2$$

$$= d^t T d$$

$$= d \cdot Td$$

$$= ||d|| \cdot ||Td|| \cdot \cos\theta_d$$

$$= ||d|| \cdot \underbrace{\alpha_d ||d||}_{\text{scaling effect of } T \text{ over } d} \cdot \cos\theta_d$$

Harris-Stephens (1988):

...calculate which directions maximize the difference

$$\sum_{(x,y)\in\mathcal{W}} [P(x,y) - P(x+d_x, y+d_y)]^2$$

$$= d^t T d$$

$$= d \cdot Td$$

$$= \|d\| \cdot \|Td\| \cdot \cos\theta_d$$

$$= \cancel{\|d\|} \cdot \alpha_d \cancel{\|d\|} \cdot \cos\theta_d$$ (unit vector)

$$= \alpha_d \cdot \cos\theta_d$$

Spectral analysis

- If λ_1 is the largest eigenvalue of T and d_1 is its corresponding unit eigenvector, then the direction d_1 gives the maximum possible value of $\alpha_\mathrm{d} \cos \theta_\mathrm{d}$ and this value is exactly λ_1.

Spectral analysis

- If λ_1 is the largest eigenvalue of T and d_1 is its corresponding unit eigenvector, then the direction d_1 gives the maximum possible value of $\alpha_{\mathrm{d}} \cos \theta_{\mathrm{d}}$ and this value is exactly λ_1.

- If λ_2 is the second eigenvalue of T and d_2 is its corresponding unit eigenvector, then the direction d_2 gives the second maximum possible value of $\alpha_{\mathrm{d}} \cos \theta_{\mathrm{d}}$ and this value is exactly λ_2.

Spectral analysis

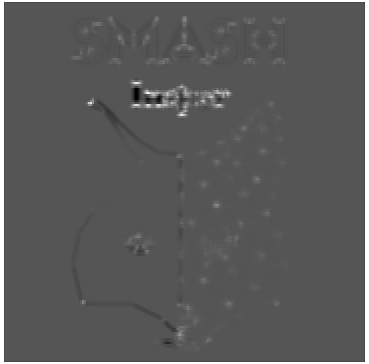
- In summary:
  - if $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, the difference is small for all directions;
  - if $\lambda_1 >> 0$ and $\lambda_2 \approx 0$, the difference is large for one direction;
  - if $\lambda_1 >> 0$ and $\lambda_2 >> 0$, the difference is large for multiple directions.

Spectral analysis

- In summary:
  - if $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, the center of the window is neither an edge or a corner;
  - if $\lambda_1 >> 0$ and $\lambda_2 \approx 0$, the center of the window is an edge;
  - if $\lambda_1 >> 0$ and $\lambda_2 >> 0$, the center of the window is a corner.

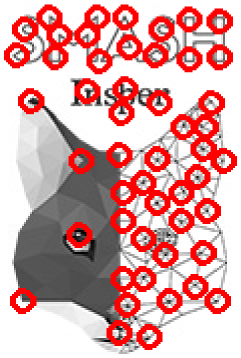
Spectral analysis

- Harris and Stephens (1988) proposed an approximation R that can be interpreted as a combined corner and edge detector:
  - if $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, then $R \approx 0$;
  - if $\lambda_1 >> 0$ and $\lambda_2 \approx 0$, then $R << 0$;
  - if $\lambda_1 >> 0$ and $\lambda_2 >> 0$, then $R >> 0$.



Spectral analysis

- Harris and Stephens (1988) proposed an approximation R that can be interpreted as a combined corner and edge detector:
  - if $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, then $R \approx 0$;
  - if $\lambda_1 >> 0$ and $\lambda_2 \approx 0$, then $R << 0$;
  - if $\lambda_1 >> 0$ and $\lambda_2 >> 0$, then $R >> 0$.
- Shi and Tomasi (1994) were only interested in corners, so they simply proposed $\min(\lambda_1, \lambda_2)$.



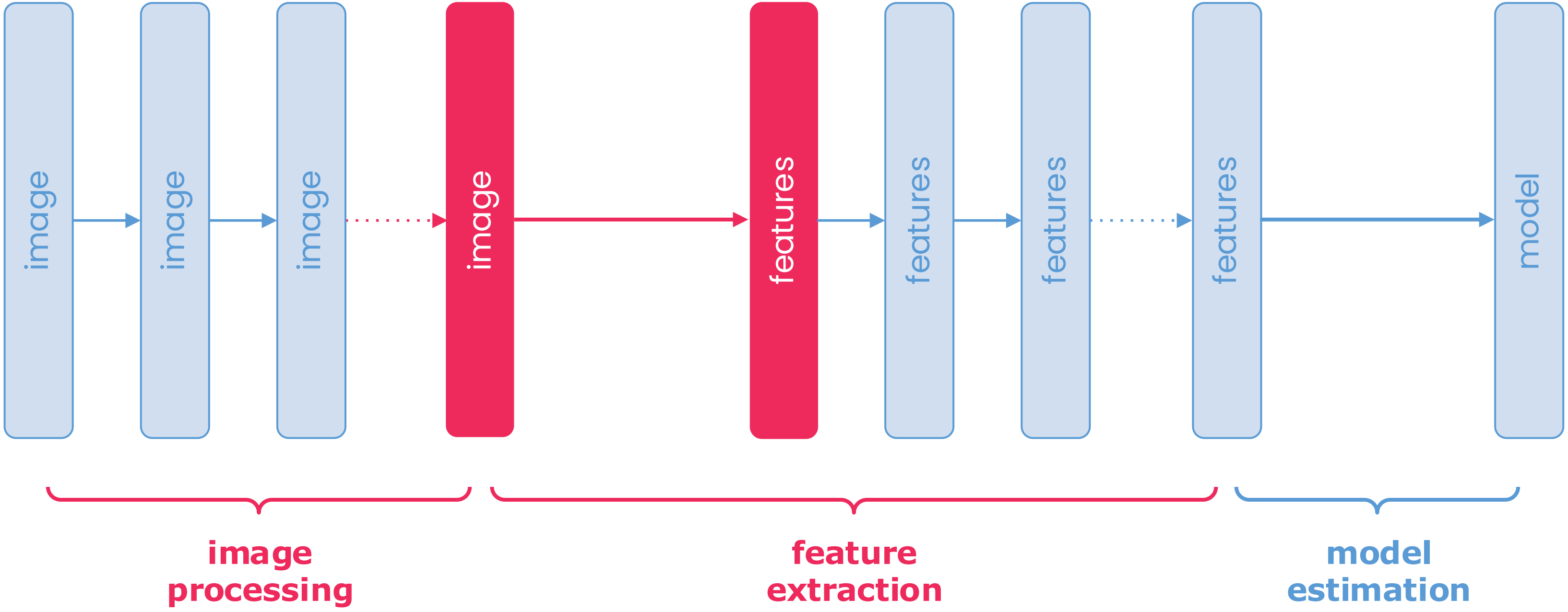


image
image
image
image
features
features
features
features
model
image
processing
feature
extraction
model
estimation

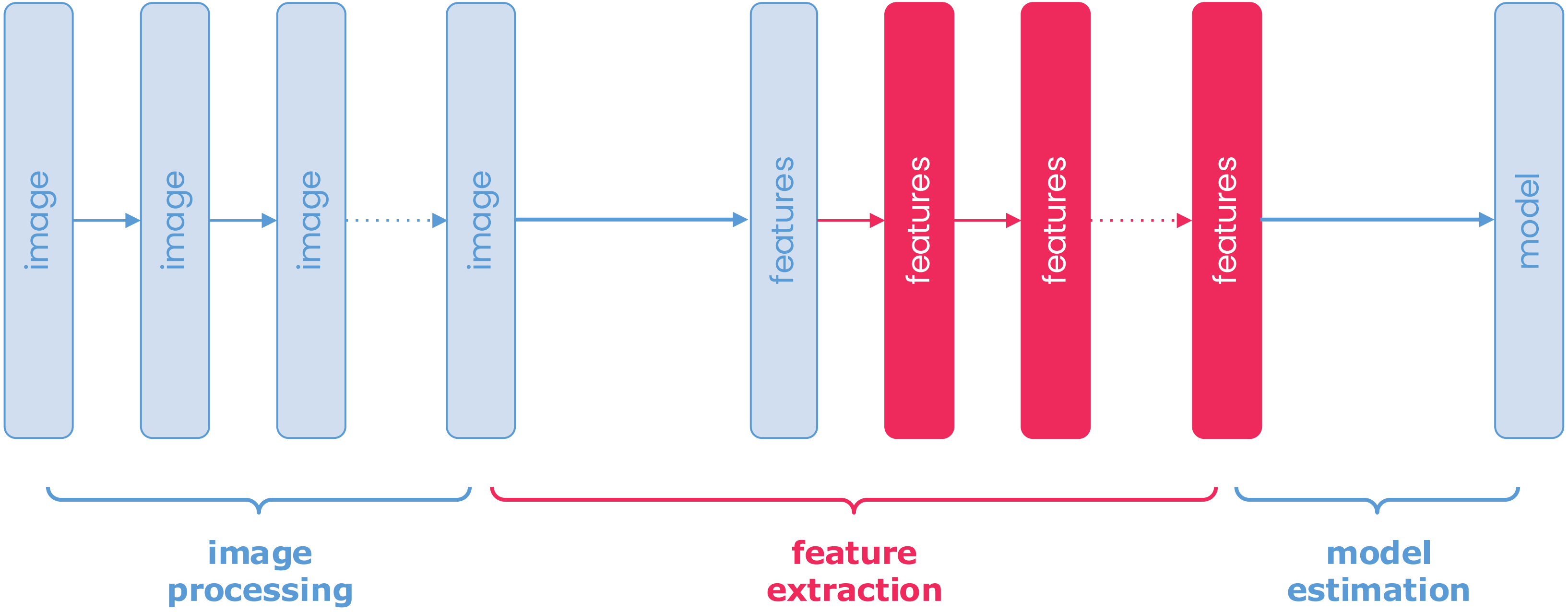


image
image
image
image
features
features
features
features
model
image processing
feature extraction
model estimation

The object detection problem

given an image of an object and a scene, detect the position of the object in the scene

The object detection problem

given an image of an object and a scene, detect the position of the object in the scene

The object detection problem

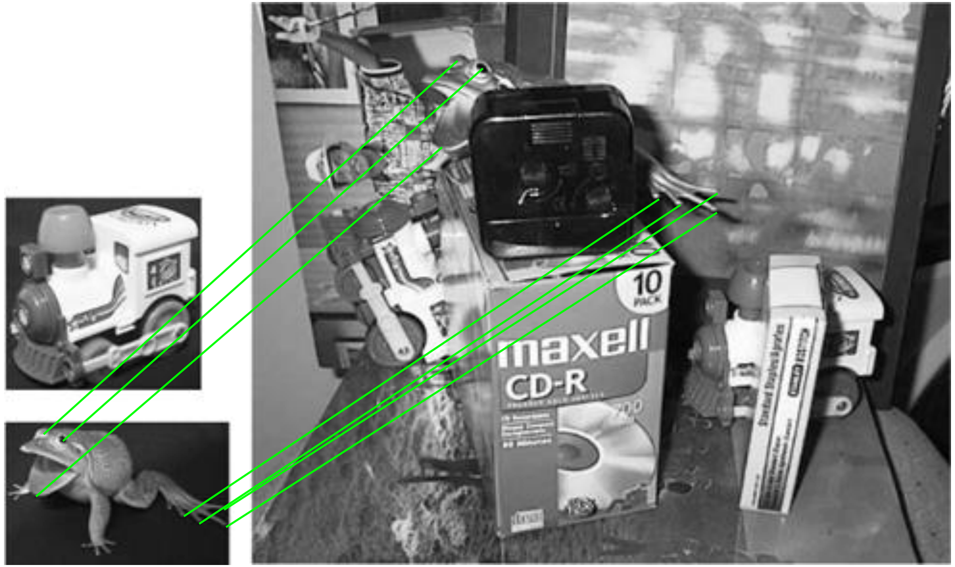
given an image of an object and a scene, detect the position of the object in the scene

The object detection problem

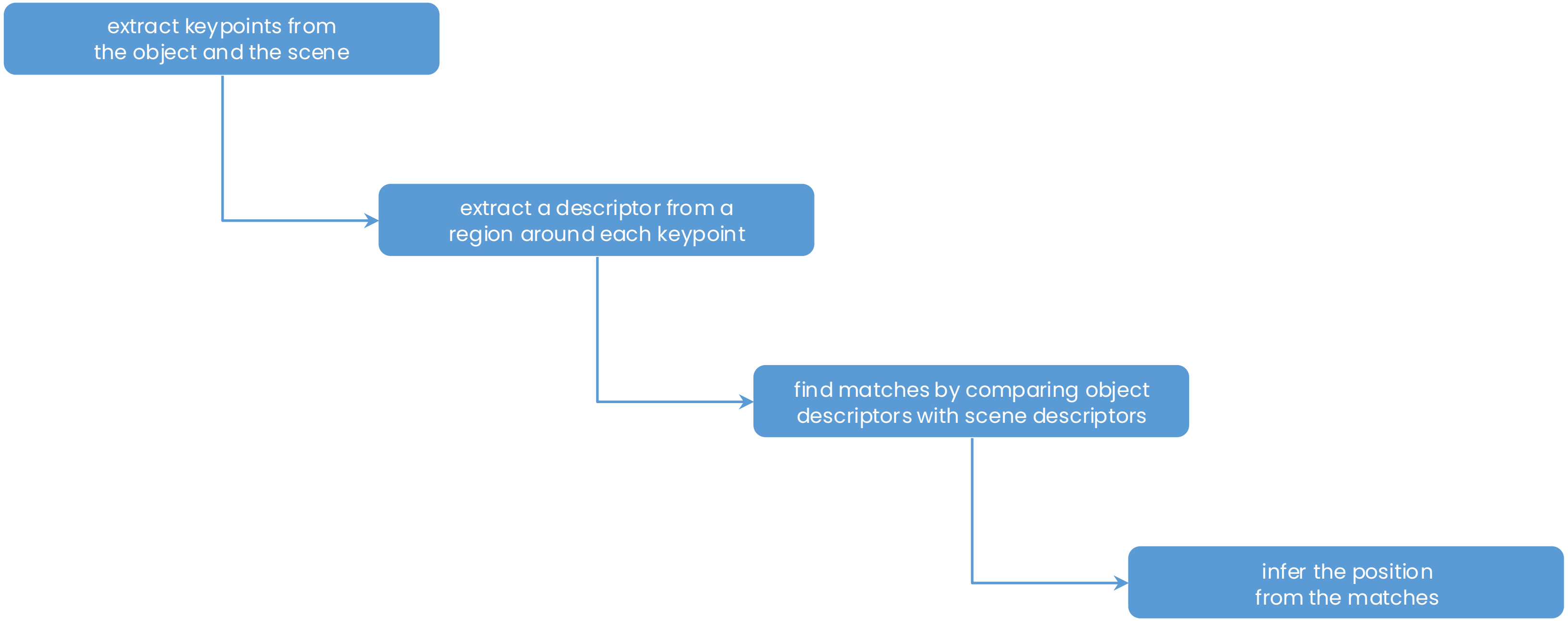
given an image of an object and a scene, detect the position of the object in the scene

The keypoint matching framework

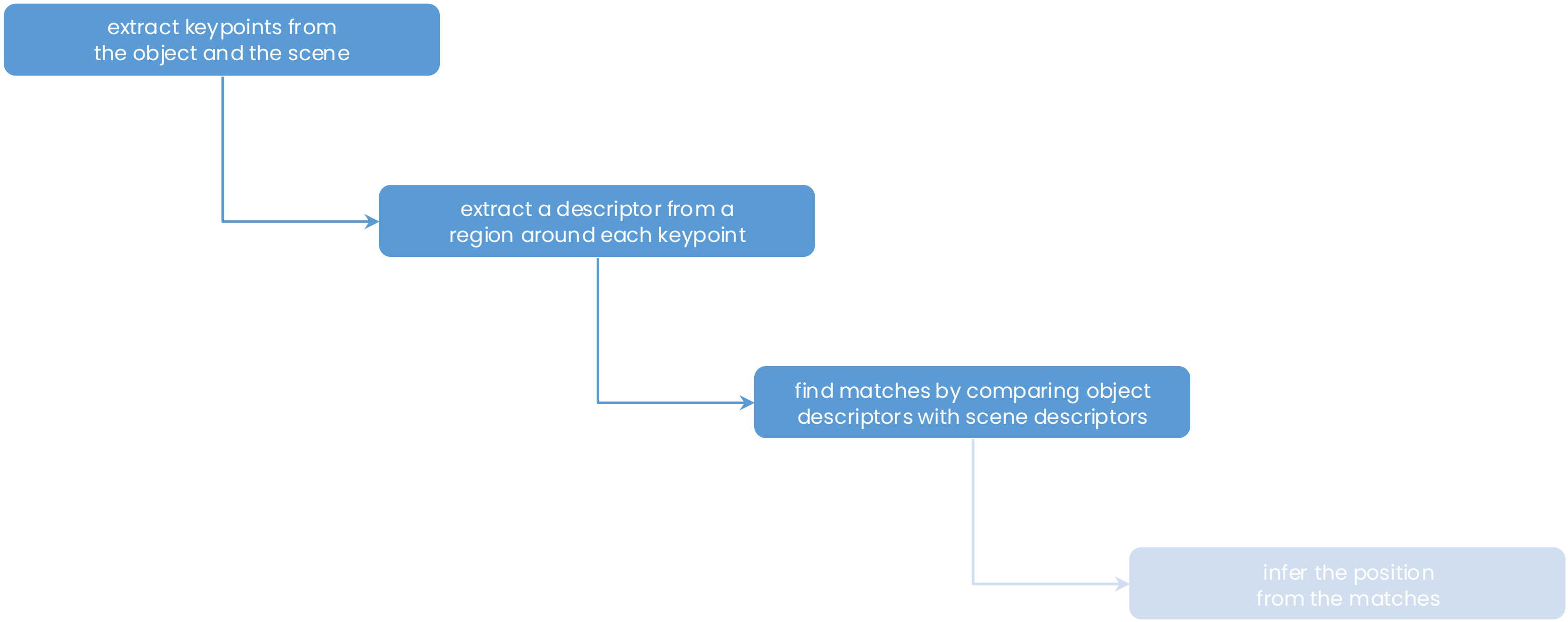
infer the position of the object from matches of keypoints *(for example, corners)*



The keypoint matching framework



The keypoint matching framework



handout

Toolkit

- **Language:** Python
- **Library:** OpenCV
- **Platform:** Google Colab

Instructions

1. Organize in groups of 2 or 3 members. No more, no less.

1. Make a copy of the notebook, read it, and do the activities.

1. Clean the notebook, save as `ipynb`, and submit via form.

Next class:

- position modeling.

Credits

This material was based on the work of other professors, listed below.

- Fabio Miranda (fabiomiranda@insper.edu.br)
- Raul Ikeda (RaulIGS@insper.edu.br)
- Fabio Ayres (FabioJA@insper.edu.br)
- Igor Montagner (IgorSM1@insper.edu.br)
- Andrew Kurauchi (AndrewTNK@insper.edu.br)
- Luciano Silva (LucianoS4@insper.edu.br)
- Tiago Sanches (tiagoss4@insper.edu.br)

Well, except for the errors. Any errors you might find are probably my fault.

Images

https://www.insper.edu.br/graduacao/organizacoes-estudantis/

Lowe, D. G. *Distinctive image features from scale-invariant keypoints.* International Journal of Computer Vision, 60. (2004)