ECE 470: Introduction to Robotics
Lecture 24

Liangjing Yang
Assistant Professor, ZJU-UIUC Institute
liangjingyang@intl.zju.edu.cn

Wechat ID: Liangjing_Yang

Overview of Robot Vision

O. Introduction to Robot Vision
- What is Robot Vision?

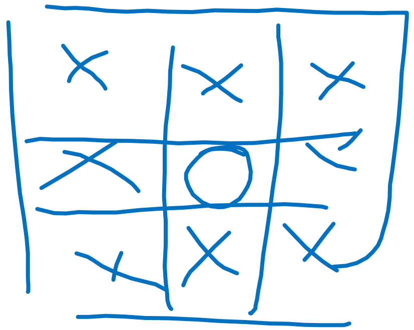
I. Image Formation
- The science behind machine vision (+ represent as a form of signal)

II. Image Processing
- **Common techniques to manipulate, enhance & analyse images**

III. Robot Vision Applications
- 3D Vision; Photogrammetry; Vision-based techniques in robotics- visual servo, pose estimation, localization, mapping, navigation

Image Processing

- Thresholding & Histogram Processing
- Filtering
- Feature Detect & Extract
  - ~~Edges & Corners~~
  - Lines, shapes, interest points



Overview of Robot Vision

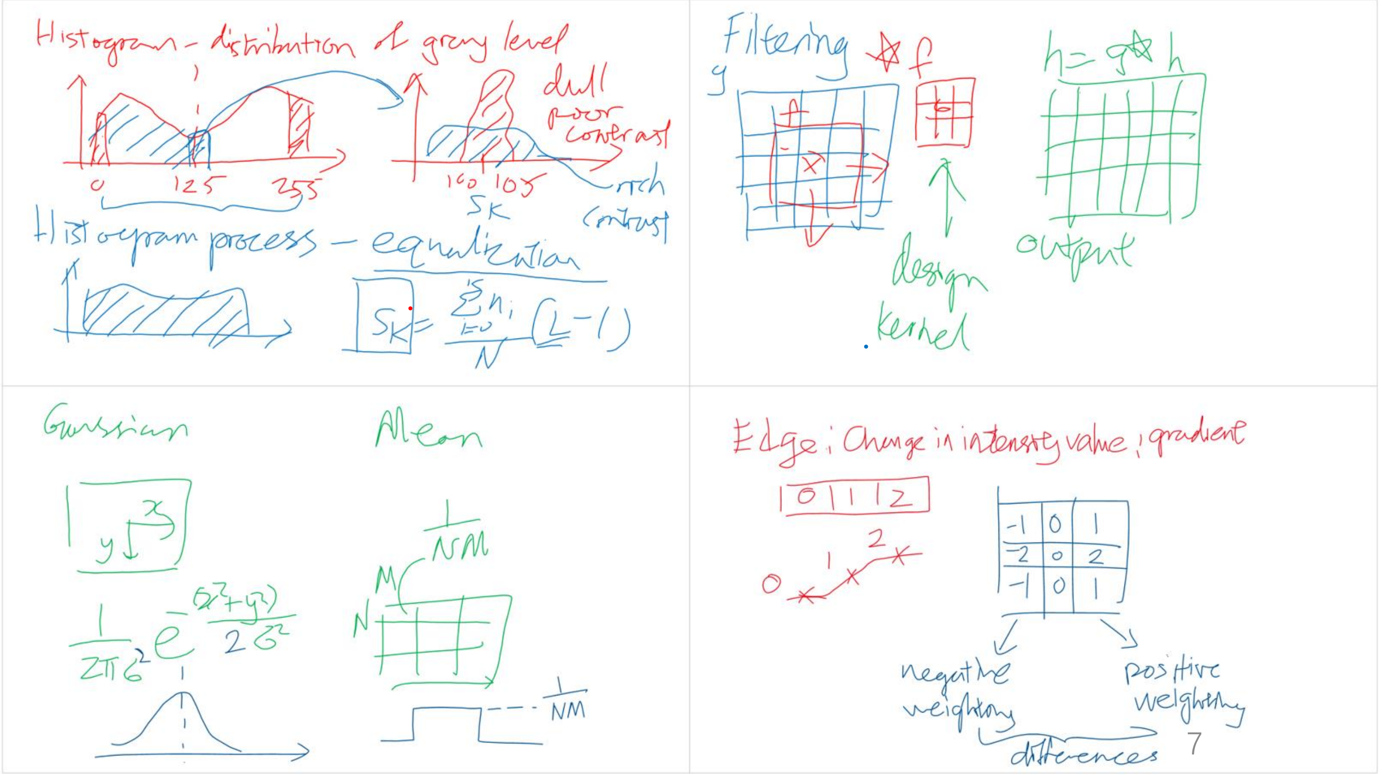
O. Introduction to Robot Vision
- What is Robot Vision?

I. Image Formation
- The science behind machine vision (+ represent as a form of signal)

II. Image Processing
- **Common techniques to manipulate, enhance & analyse images**

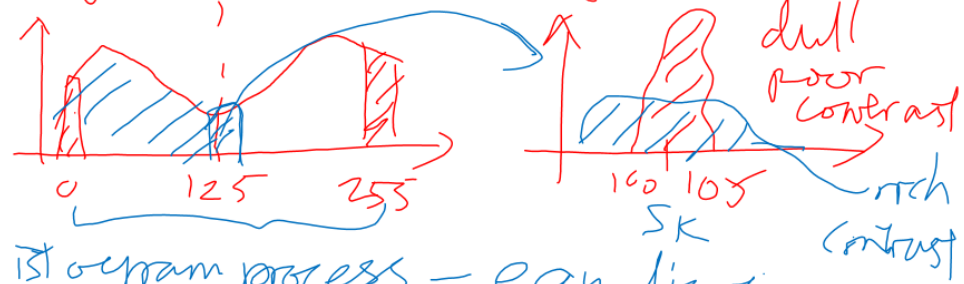
III. Robot Vision Applications
- 3D Vision; Photogrammetry; Vision-based techniques in robotics- visual servo, pose estimation, localization, mapping, navigation

Draft



Draft

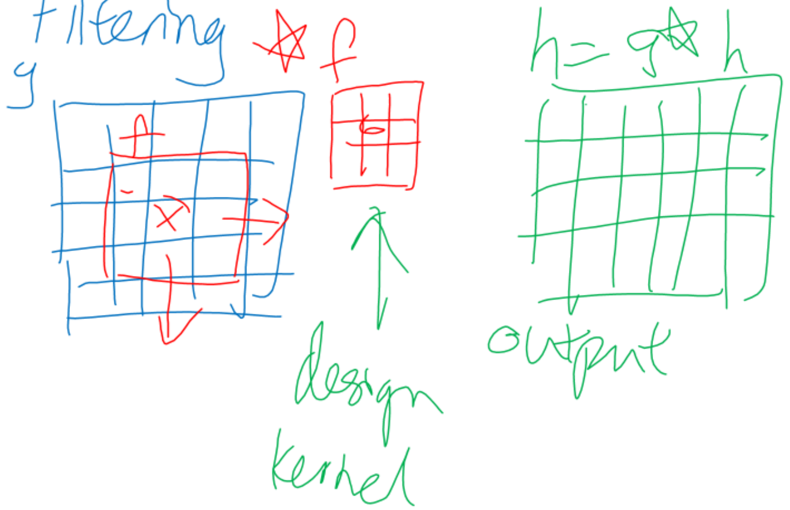
Histogram - distribution of grey level



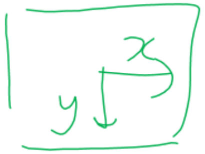
Histogram process - equalization

$$\boxed{S_K} = \frac{\sum_{i=0}^{K} n_i}{N}(L-1)$$

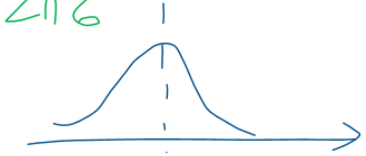
Filtering $g \star f$



Gaussian



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



Mean

$$\frac{1}{NM}$$

Edge: Change in intensity value; gradient

0	1	1	2

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

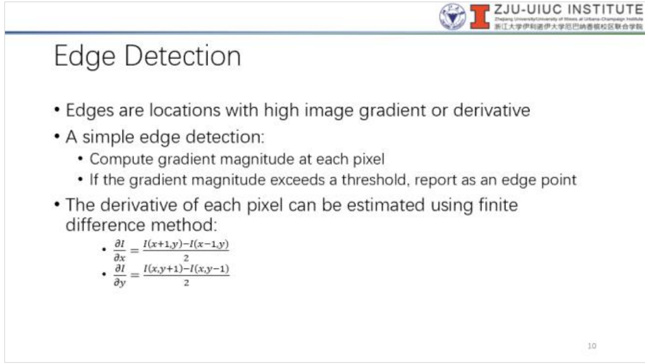
negative weighting

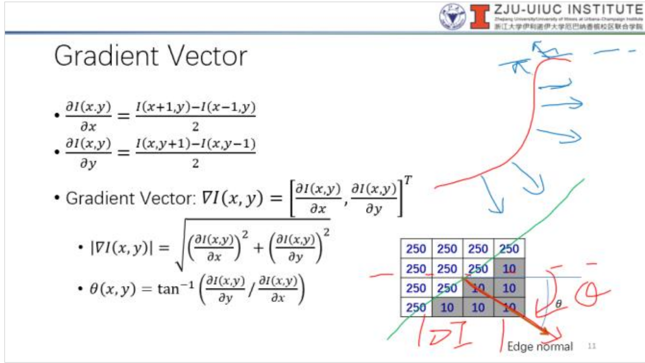
positive weighting

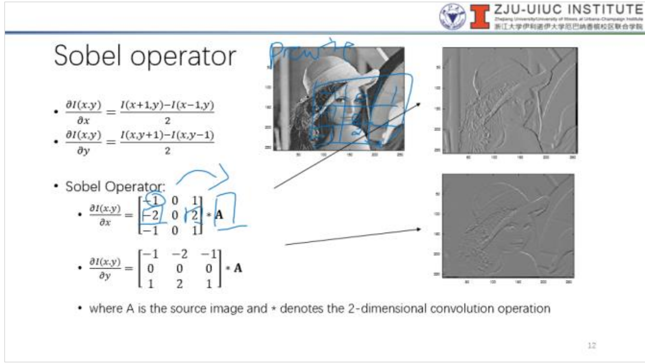
differences

Image Processing

- Thresholding & Histogram Processing
- Filtering
- Feature Detect & Extract
    - Edges & Corners
    - Lines, shapes, interest points







Canny Edge Detection (…… Last Lecture)

- Canny edge detection is probably the most used and taught edge detection algorithm
- Involves 5 steps:
  1. Apply Gaussian filter to smoothen the image in order to remove the noise
  2. Find the intensity gradients of the image
  3. Apply non-maximum suppression to get rid of spurious response to edge detection
  4. Apply edge detection using two threshold value
  5. Finalize edge detection by hysteresis
- J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986.

Canny Edge Detection

- Step (1): Apply Gaussian filter

Canny Edge Detection

- Step (1): Apply Gaussian filter to smooth the image in order to remove the noise
  - Edge detection are easily affected by image noise
  - A Gaussian filter is applied to convolve with the image
    - $h(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$
  - This will smooth the image to reduce the effects of obvious noise on the edge detector
- Question: Does this give you a sharper image?

Not this step

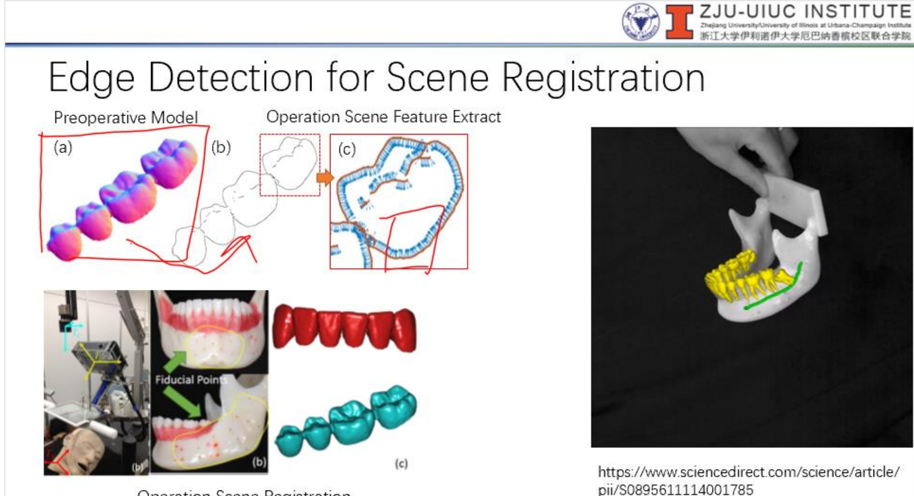
Canny Edge Detection

- Step (2): Find the intensity gradients of the image

Canny Edge Detection

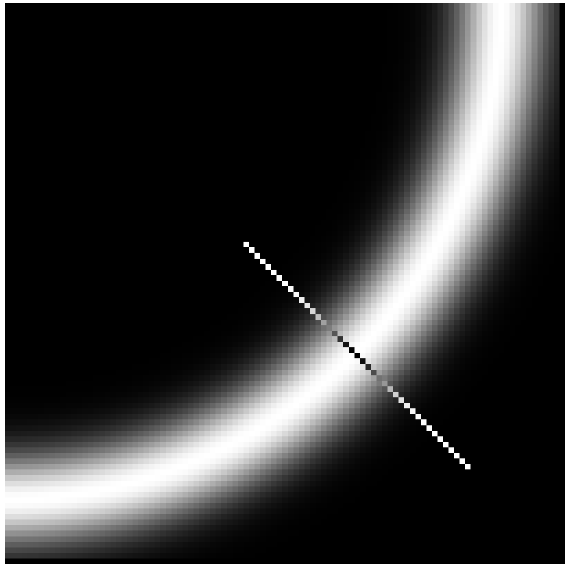
- Step (2): Find the intensity gradients of the image
  - $\frac{\partial I(x.y)}{\partial x} = \frac{I(x+1,y)-I(x-1,y)}{2}$
  - $\frac{\partial I(x,y)}{\partial y} = \frac{I(x,y+1)-I(x,y-1)}{2}$
  - Gradient Vector: $\nabla I(x,y) = \left[\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right]^T$
    - $|\nabla I(x,y)| = \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2}$
    - $\theta(x,y) = \tan^{-1}\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right)$

Application Example



Canny Edge Detection

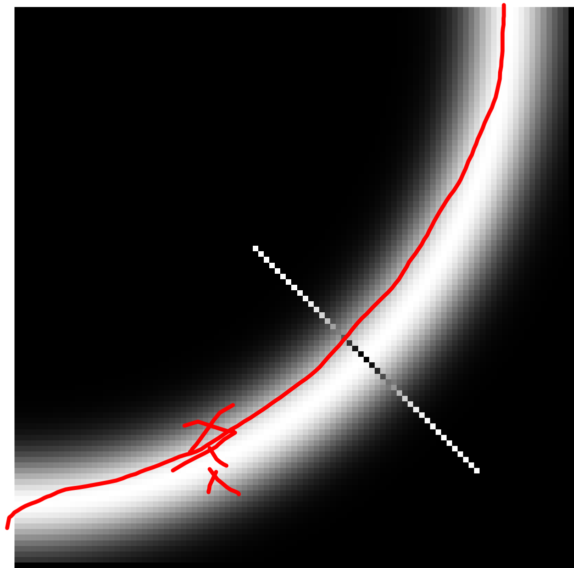
- Step (2): Find the intensity gradients of the image
  - $\frac{\partial I(x.y)}{\partial x} = \frac{I(x+1,y)-I(x-1,y)}{2}$ Δ in x-direction
  - $\frac{\partial I(x,y)}{\partial y} = \frac{I(x,y+1)-I(x,y-1)}{2}$ 〃 〃 y- 〃
  - Gradient Vector: $\nabla I(x,y) = \left[\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right]^T$
    - $|\nabla I(x,y)| = \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2}$
    - $\theta(x,y) = \tan^{-1}\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right)$



Canny Edge Detection

- How do we precisely localize the edge?

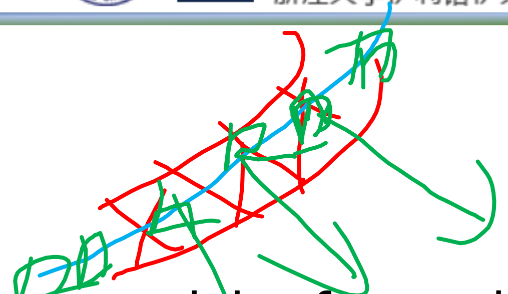
Thinning of the edge



Canny Edge Detection

- Step (3): Apply non-maximum suppression to get rid of spurious response to edge detection
- Non-maximum suppression is an edge thinning technique

Canny Edge Detection



- Step (3): Apply non-maximum suppression to get rid of spurious response to edge detection
- Non-maximum suppression is an edge thinning technique
  - Close to the center of the true edge
  - non-maximum suppression can help to suppress all the gradient values to 0 except the local maximal, which indicates location with the sharpest change of intensity value
  - Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions
  - If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction, the value will be preserved. Otherwise, the value will be suppressed.

Canny Edge Detection

- Step (4): Apply edge detection using two threshold value K_H and K_L
- $Edge(x,y) = \begin{cases} E_{strong} & if \ \ |\nabla I(x,y)| > K_H \\ E_{average} & if \ \ K_L \leq |\nabla I(x,y)| \leq K_H \\ E_{weak} & if \ \ |\nabla I(x,y)| < K_L \end{cases}$

Canny Edge Detection

- Step (4): Apply edge detection using two threshold value K_H and K_L
- $Edge(x,y) = \begin{cases} E_{strong} & if\ \ |\nabla I(x,y)| > K_H \\ E_{average} & if\ \ K_L \le |\nabla I(x,y)| \le K_H \\ E_{weak} & if\ \ |\nabla I(x,y)| < K_L \end{cases}$

→ definitely edge

→ definitely Not

Canny Edge Detection

- Step (4): Apply edge detection using two threshold value K_H and K_L
- $$Edge(x,y) = \begin{cases} E_{strong} & if \ |\nabla I(x,y)| > K_H \\ E_{average} & if \ \ K_L \le |\nabla I(x,y)| \le K_H \\ E_{weak} & if \ |\nabla I(x,y)| < K_L \end{cases}$$

✓ marginal ✓

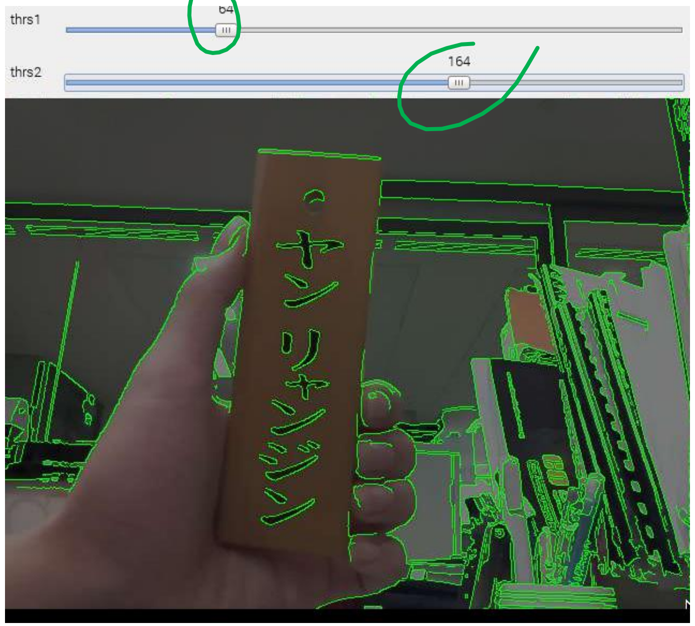
$\underbrace{\quad E_{weak} \quad}$ $\underbrace{\quad E_{average} \quad}$ $\underbrace{\quad E_{strong} \quad}$

K_L

K_H

decided by developer (user)

edge.py
cv2.Canny(image, thrs1, thrs2, size)



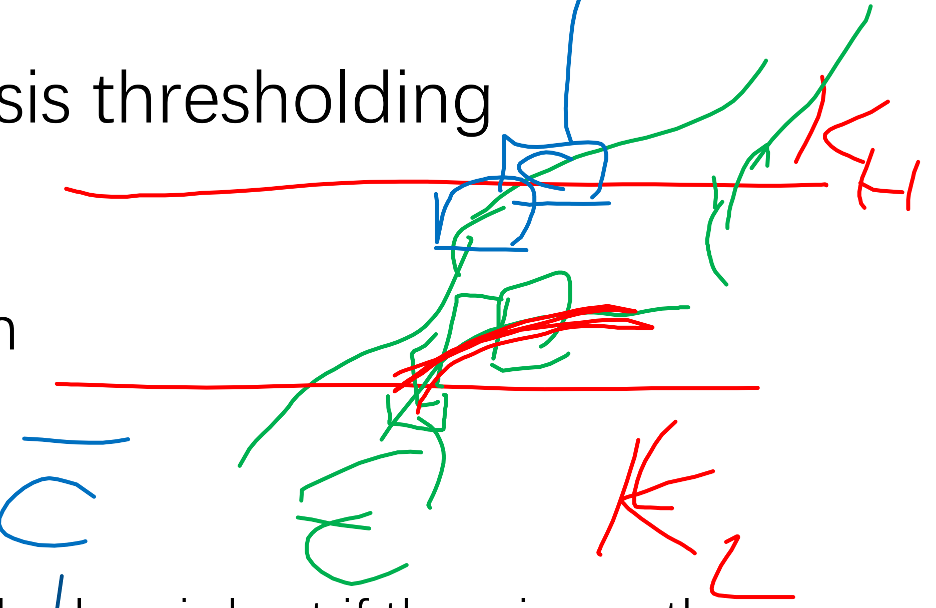
Canny Edge Detection

- Step (5): Finalize edge detection by hysteresis thresholding
  - Small K means more details
  - High K includes more noise
  - Hysteresis Thresholding allows us to apply both
    - Keep both high threshold K_H and low threshold K_L
    - Any edges with magnitude < K_L are discarded
    - Any edges with magnitude > K_H are kept
    - An edge with magnitude between the two threshold values is kept if there is a path of edges with magnitude > K_L connecting the edge to another edge with magnitude > K_H

Canny Edge Detection

C ' confirmed cases

- Step (5): Finalize edge detection by hysteresis thresholding
  - Small K means more details
  - High K includes more noise
  - Hysteresis Thresholding allows us to apply both
    - Keep both high threshold K_H and low threshold K_L
    - Any edges with magnitude < K_L are discarded
    - Any edges with magnitude > K_H are kept
    - An edge with magnitude between the two threshold values is kept if there is a path of edges with magnitude > K_L connecting the edge to another edge with magnitude > K_H



confirmed Not cases

Edge Detection

Original ------------ ------------

Edge Detection

discontinuous

continuous edge

Image Processing

- Thresholding & Histogram Processing
- Filtering
- Feature Detect & Extract
  - Edges & Corners
  - Lines, shapes, interest points

Corner Detection

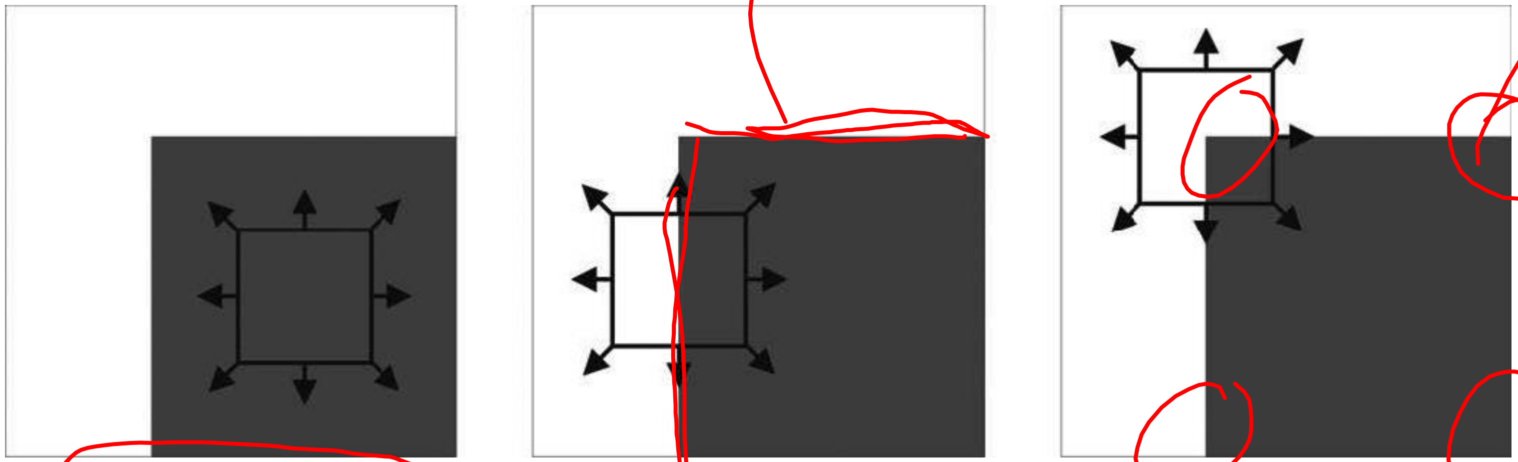
Interest pt

- Corner detection are used for many image feature extraction
  - Because corners are features with high repeatability

edge

corner

blob



Flat region

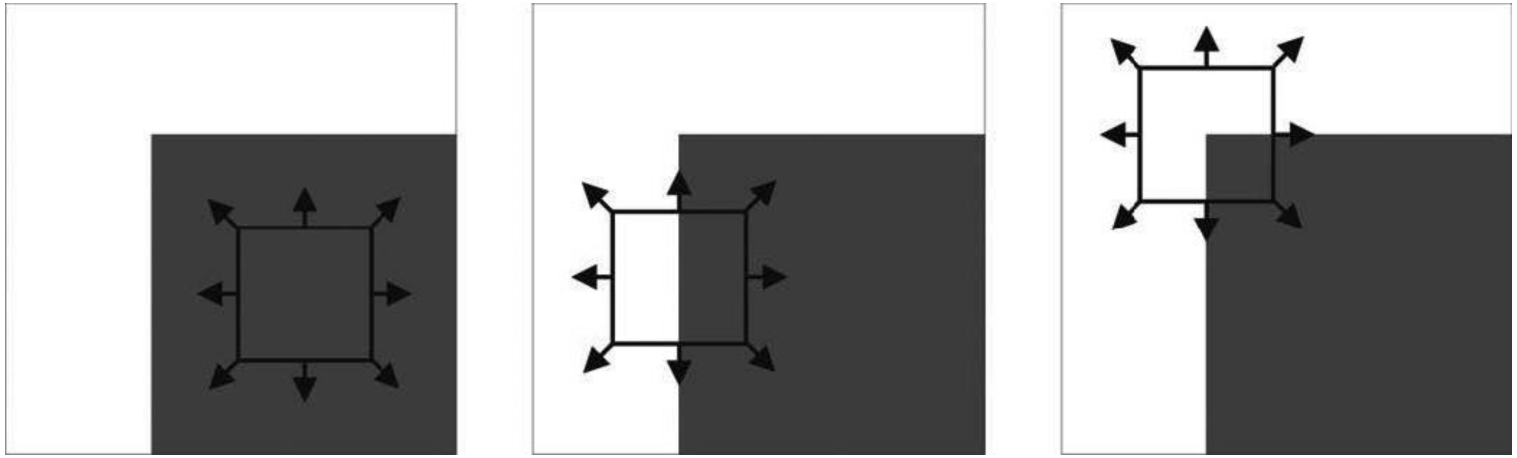
Edge

Corner

Images taken from Introduction to Autonomous Mobile Robot

Corner Detection

- Basic idea of corner detection: large change in appearance
  - Flat region: no change
  - Edge: no change along edge
  - Corner: significant change in more than one direction



Images taken from Introduction to Autonomous Mobile Robot

Harris Corner Detector

- Consider taking an image patched centered on $(u, v\ )$ and shifting it by (x, y), the sum of square differences SSD between these two patches is:

$$SSD(x, y) = \sum_{u} \sum_{v} [I(u, v) - I(u + x, v + y)]^2$$

- Using first-order Talyor expansion,

$$I(u + x, v + y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y$$

- Hence, SSD becomes

$$SSD(x, y) \approx \sum_{u} \sum_{v} \left[I_x(u, v)x + I_y(u, v)y\right]^2$$

$$= \sum_{u} \sum_{v} \left[I_x^2 x^2 + 2xyI_xI_y + I_y^2 y^2\right]^2$$

Harris Corner Detector

$$SSD(x,y) = \sum_u \sum_v \left[I_x^2 x^2 + 2xyI_xI_y + I_y^2 y^2\right]^2$$
$$= \sum_u \sum_v \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
$$= \begin{bmatrix} x & y \end{bmatrix} \sum_u \sum_v \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- $SSD(x,y) = \begin{bmatrix} x & y \end{bmatrix} M \begin{bmatrix} x \\ y \end{bmatrix}$
- Since M is symmetric, we can rewrite the matrix as:

$$M = A^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} A$$

  - where λ_1 and λ_2 are the eigenvalues of M

Harris Corner Detector

$$SSD(x,y) = \sum_u \sum_v [I_x^2 x^2 + 2xyI_xI_y + I_y^2 y^2]^2$$
$$= \sum_u \sum_v [x \quad y] \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
$$= [x \quad y] \underbrace{\sum_u \sum_v \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}}_{\text{structural tensor}} \begin{bmatrix} x \\ y \end{bmatrix}$$

- $SSD(x,y) = [x \quad y] M \begin{bmatrix} x \\ y \end{bmatrix}$
- Since M is symmetric, we can rewrite the matrix as:

$$M = A^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} A$$

  - where λ_1 and λ_2 are the eigenvalues of M

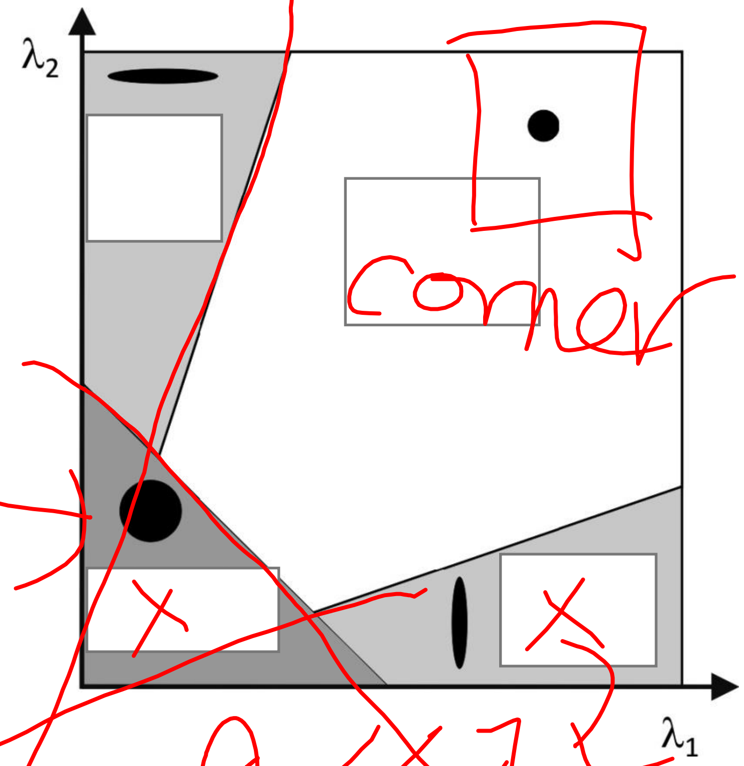
Edge in δ_2

Harris Corner Detector

flat region
wide

- As mentioned, a corner is characterized by a large variation of SSD in all direction, the larger the variation in that direction

- Both λ are small means flat region
- One strong and one weak λ means edge
- Two strong λ means corner

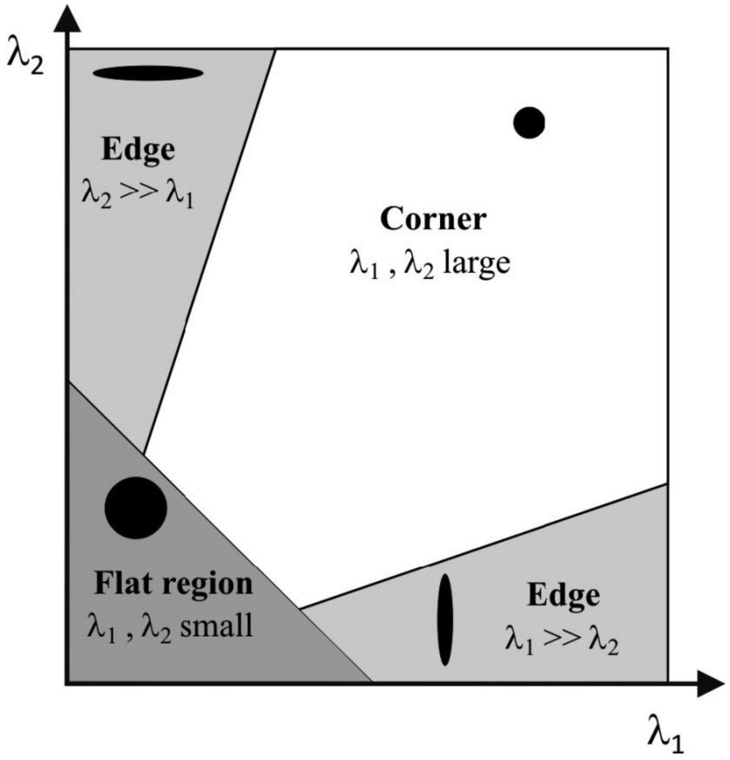


$\lambda_2 << \lambda_1$

Harris Corner Detector

- As mentioned, a corner is characterized by a large variation of SSD in all direction, the larger the variation in that direction

- Both λ are small means flat region
- One strong and one weak λ means edge
- Two strong λ means corner

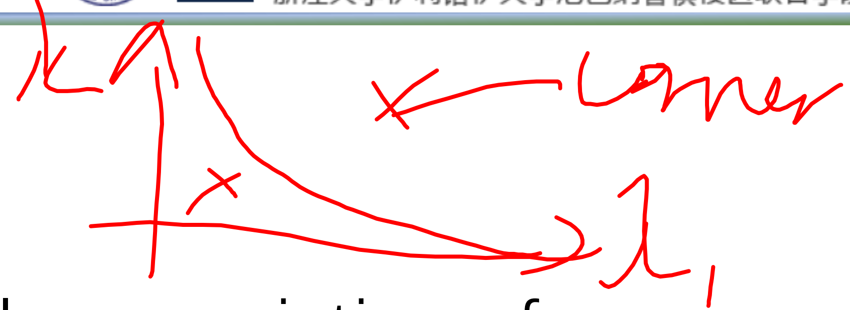


Quick way of Calculating Corner Response:

$$R = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 = \det(M) - k \cdot \mathrm{tr}(M)^2$$

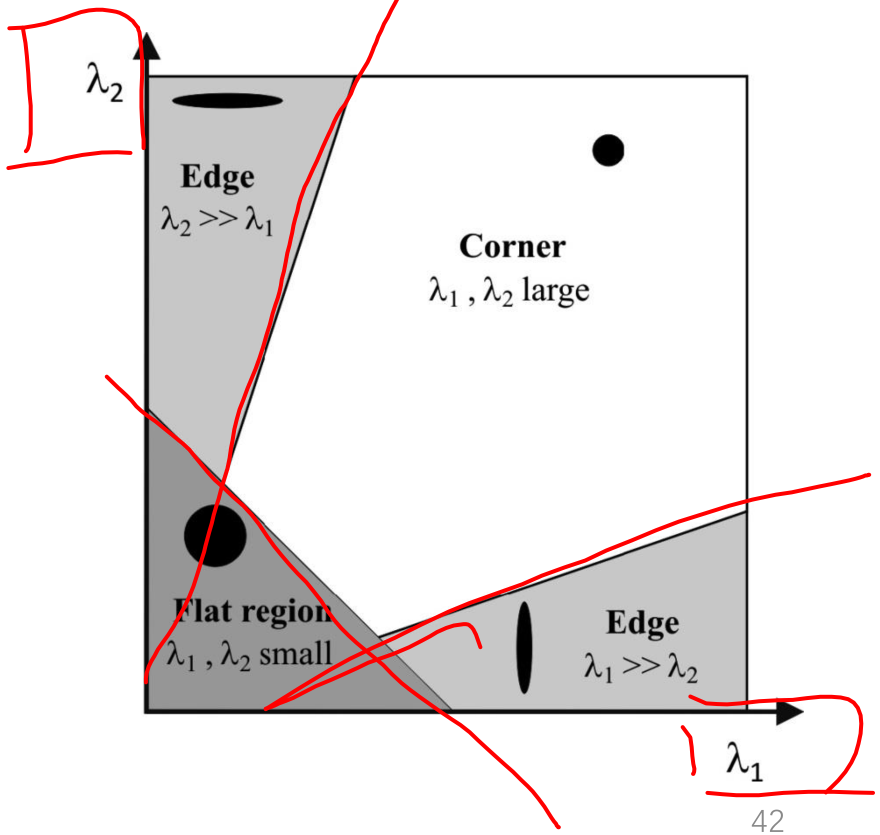
where k is an empirically determined constant; $k \in [0.04, 0.06]$.

Harris Corner Detector



- As mentioned, a corner is characterized by a large variation of SSD in all direction, the larger the variation in that direction

- Both λ are small means flat region
- One strong and one weak λ means edge
- Two strong λ means corner



Quick way of Calculating Corner Response:

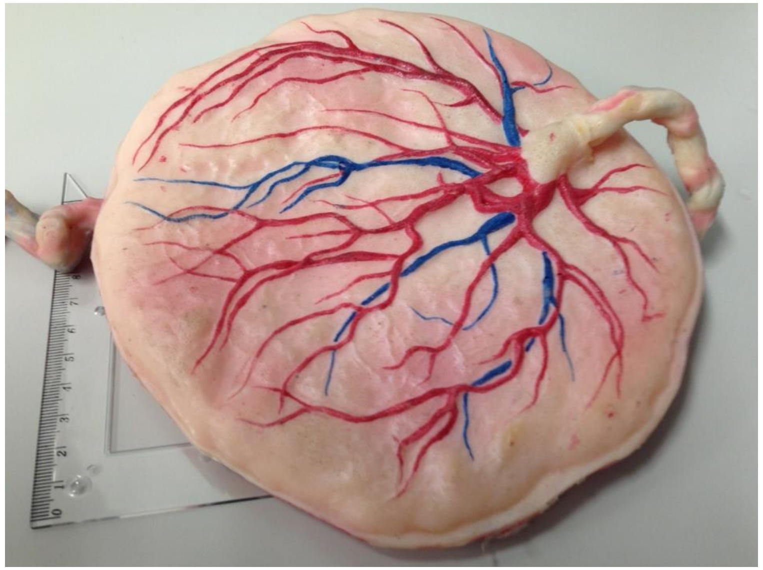
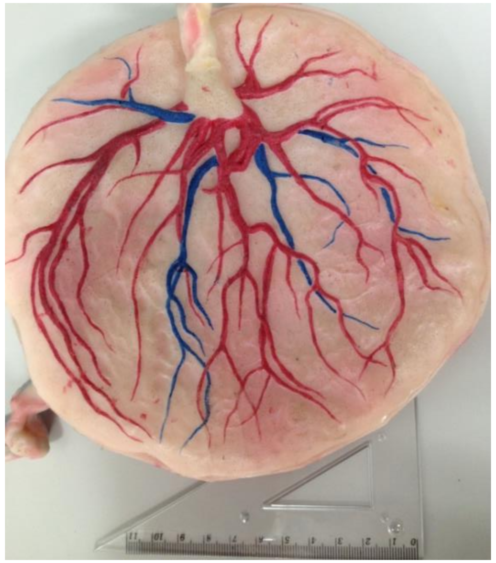
trace

$$R = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 = \det(M) - k \cdot \mathrm{tr}(M)^2$$

where k is an empirically determined constant; $k \in [0.04, 0.06]$.

Match two images

- Are these objects the same?
  - Pixel to pixel comparison might not work

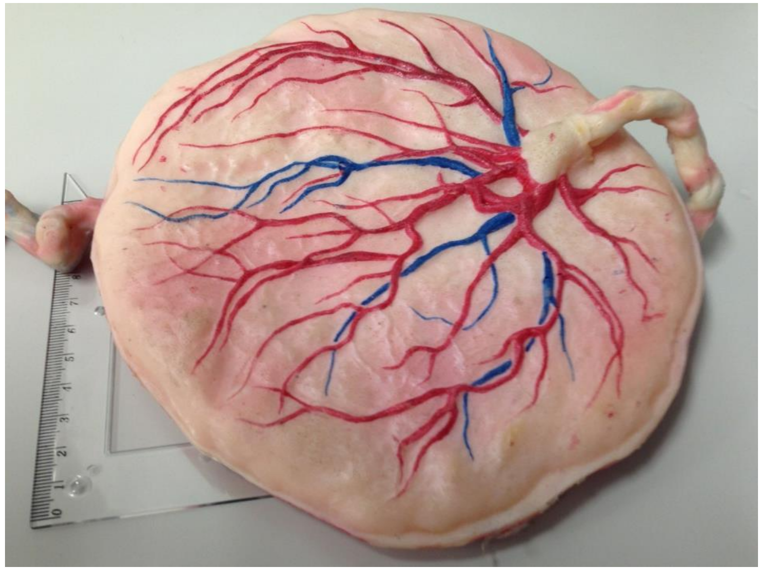



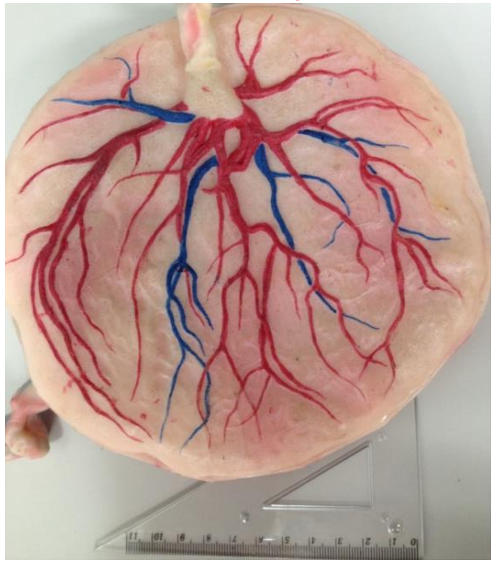

Match two images

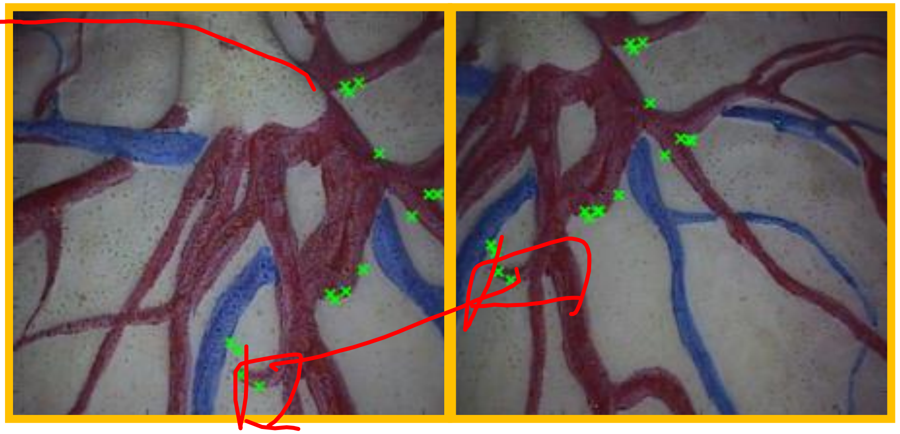
Feature detected can be interest pt (Feature Descriptor)

- Are these objects the same?
  - Pixel to pixel comparison might not work

Descriptor

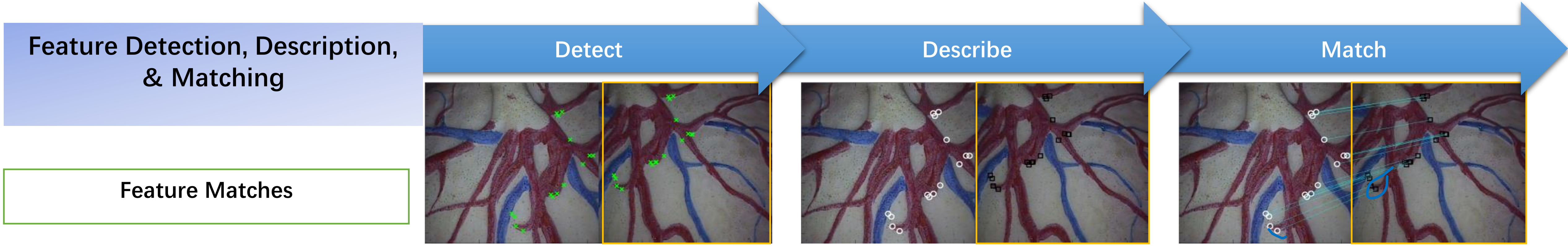






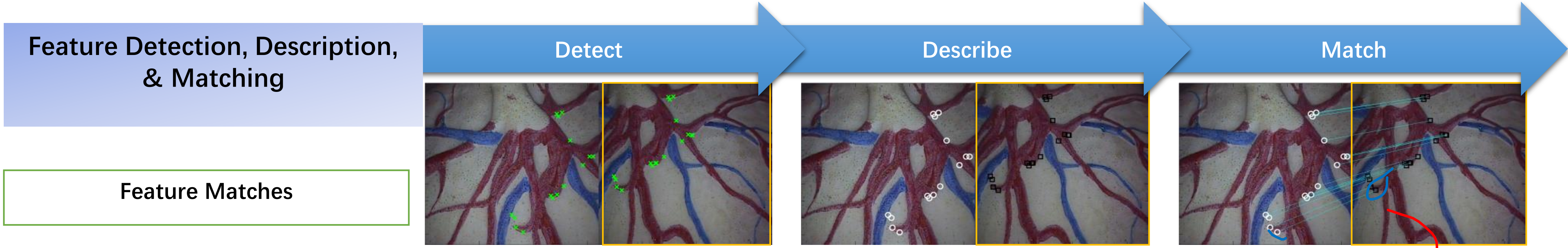
Match two images

- Feature matching
- Detect; Describe; Match; Transform (for image mapping)

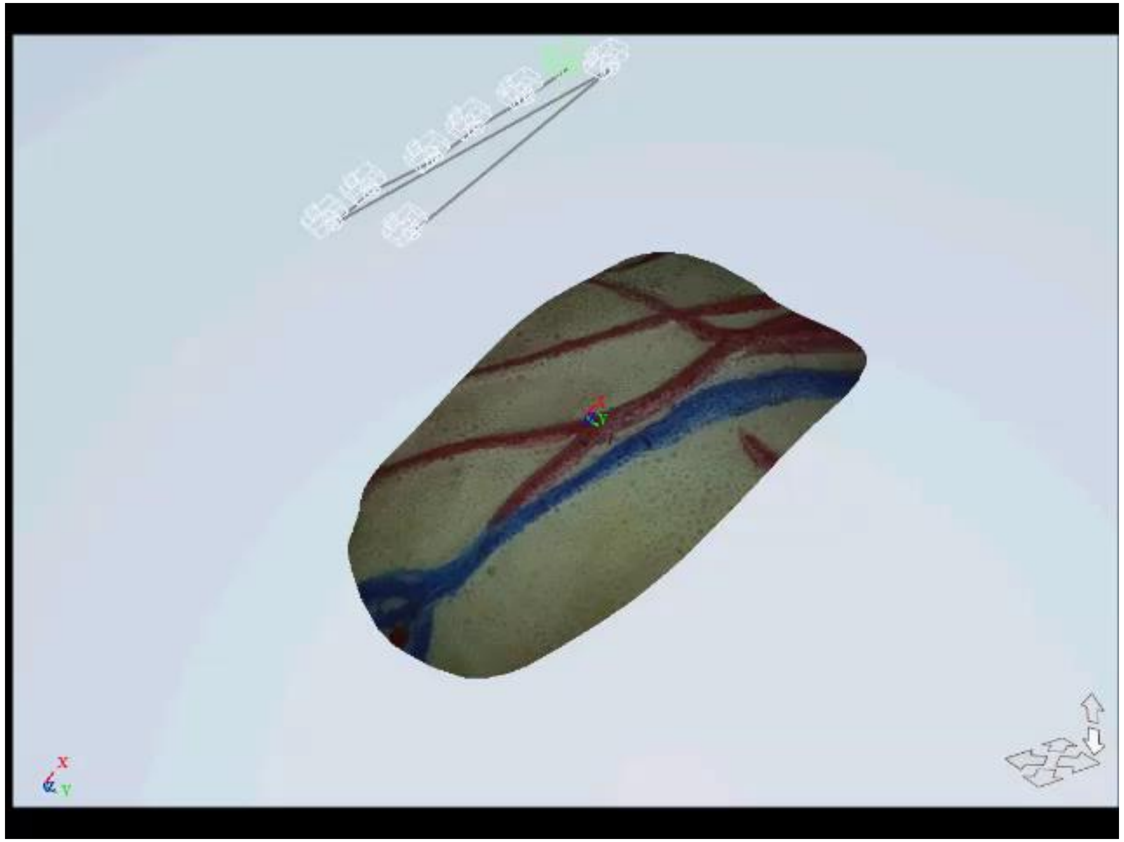


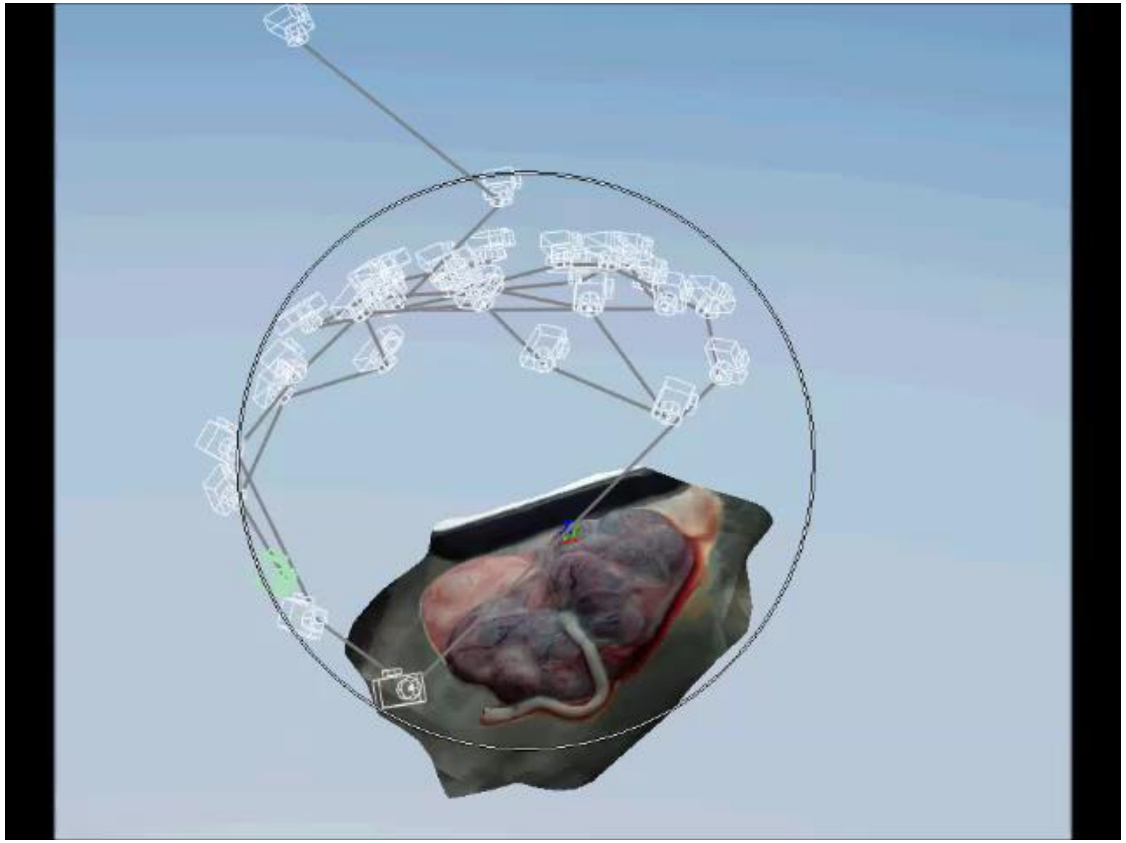
Match two images

- Feature matching
- Detect; Describe; Match; Transform (for image mapping)



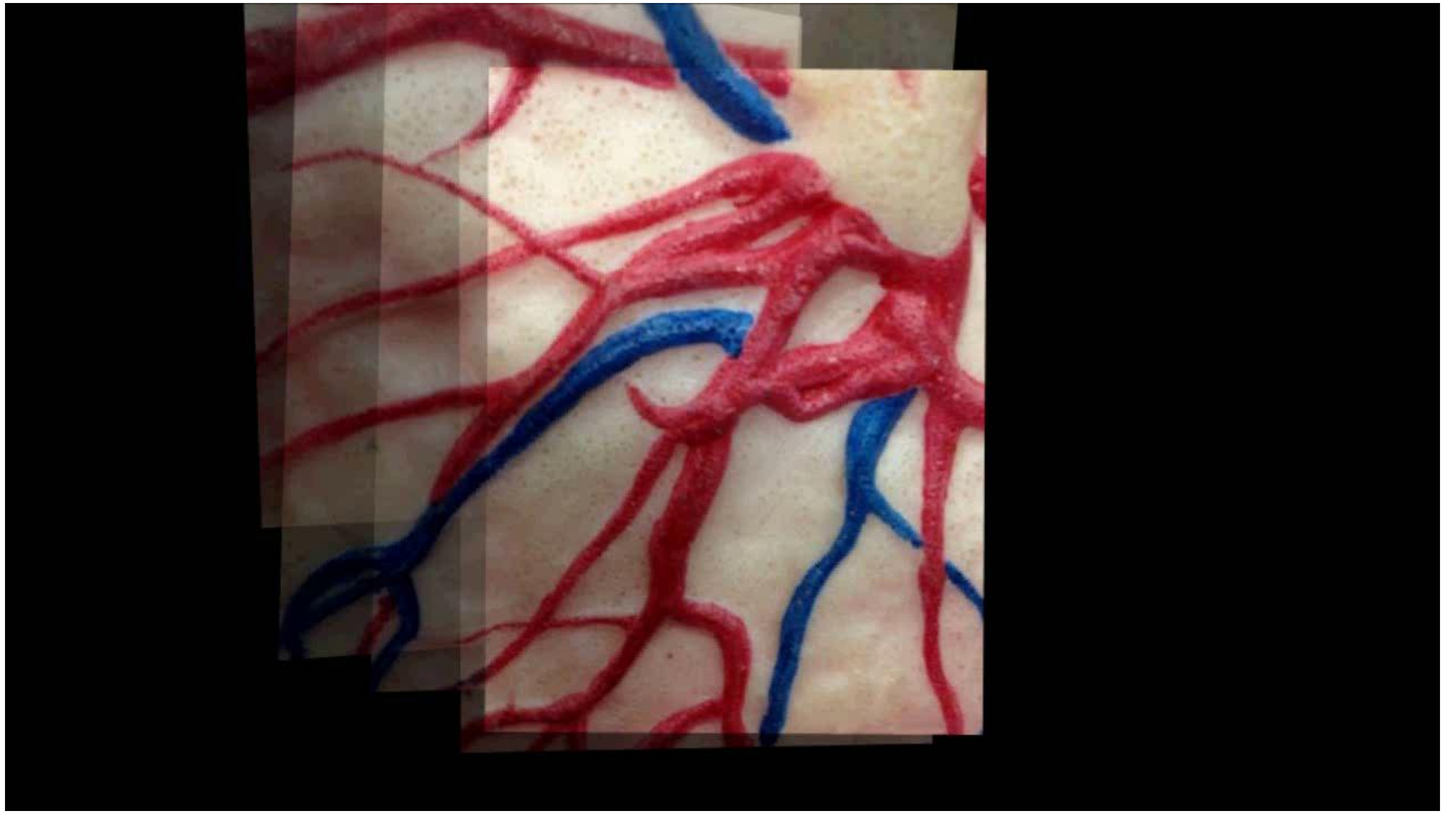
Example of Applications in Corner Detection



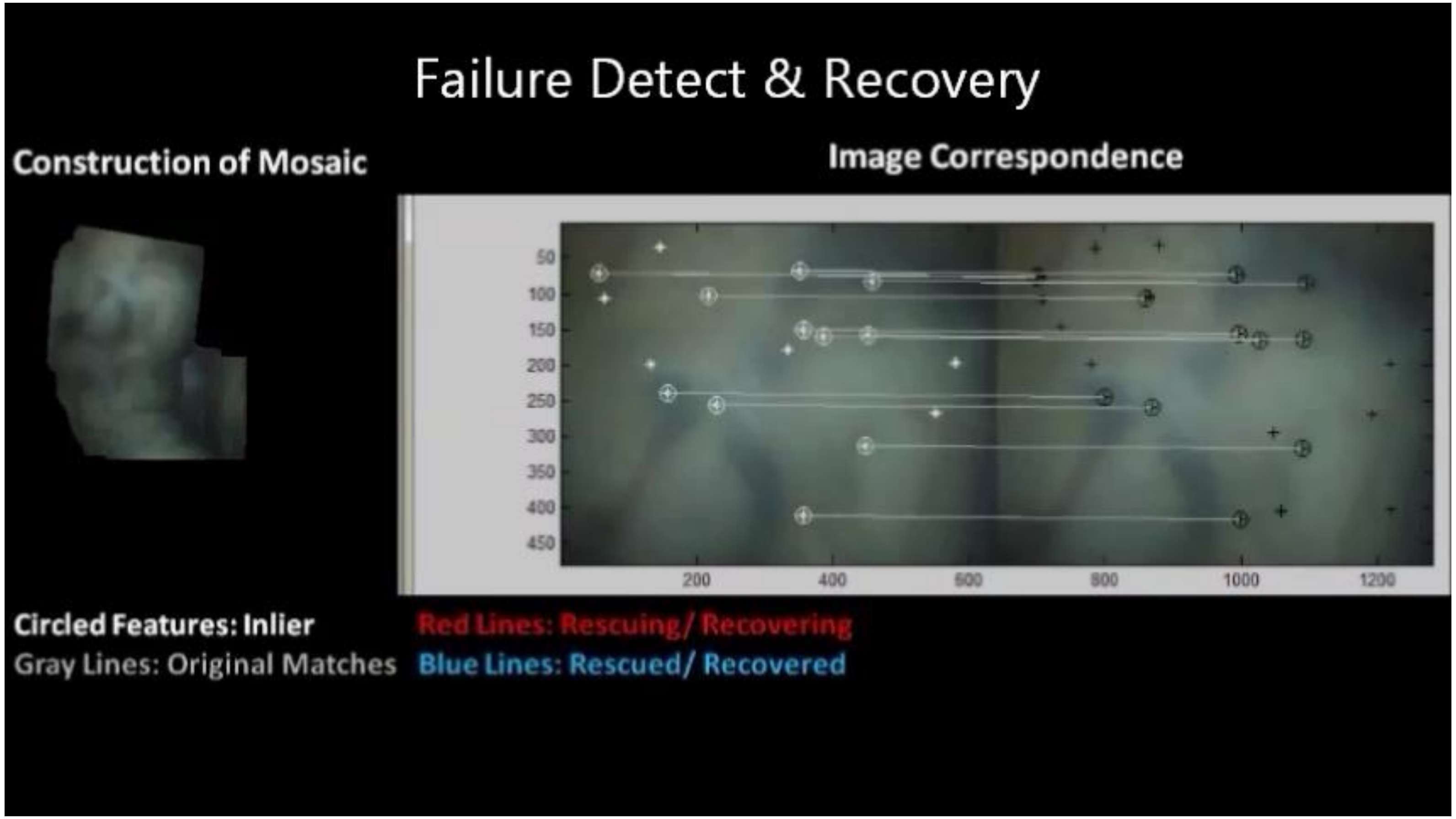


Example of Applications in Corner Detection

- Microsoft Photosynth



Dealing with Failure



Yang, L., et al. (2015). "Towards scene adaptive image correspondence for placental vasculature mosaic in computer assisted fetoscopic procedures." *Int J of Med Robot* **12**(3): 375--386.

Image Processing

- Image Enhancement
  - ☑Thresholding & Histogram Processing
  - ☑ Filtering
- Image Analysis
  - ☑ Feature Detection
    - ☑ Edges
    - ☑ Interest points- Corners
    - Lines & Shapes
    - Target Tracking