



COMPUTER VISION

Locating Interesting Points

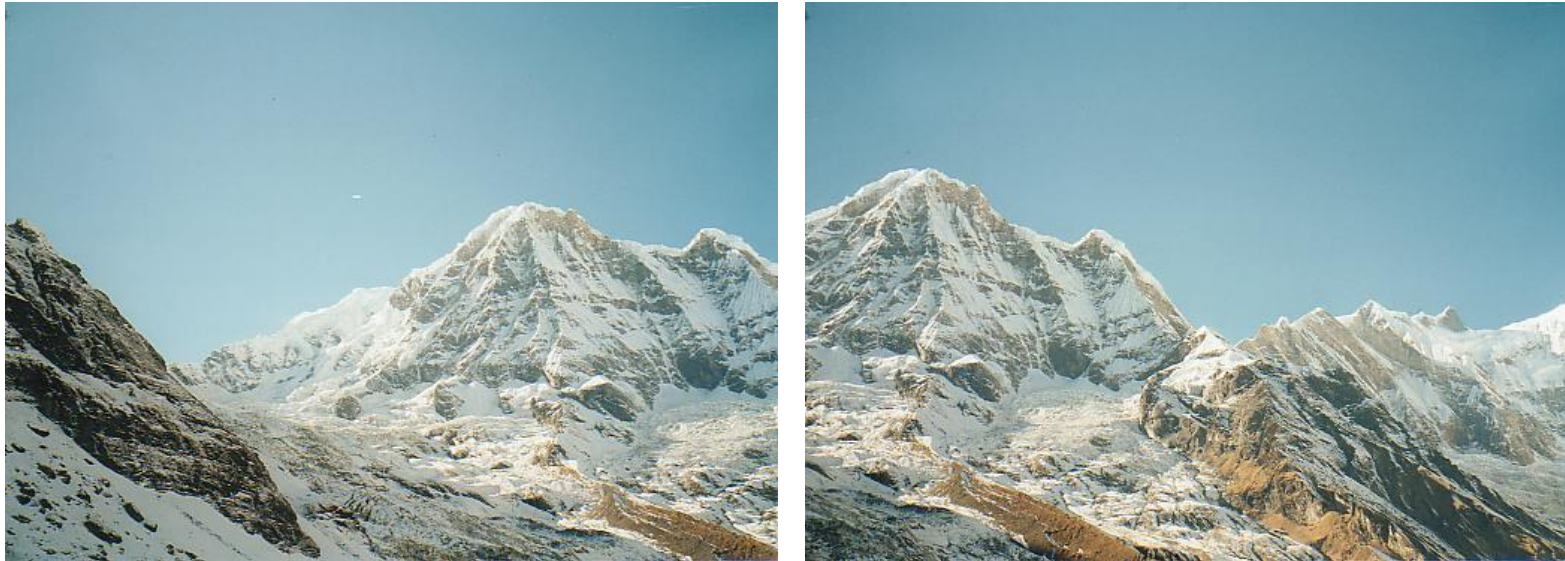
Le Thanh Ha, Ph.D

Assoc. Prof. at University of Engineering and Technology,
Vietnam National University

ltha@vnu.edu.vn; lthavnu@gmail.com; 0983 692 592

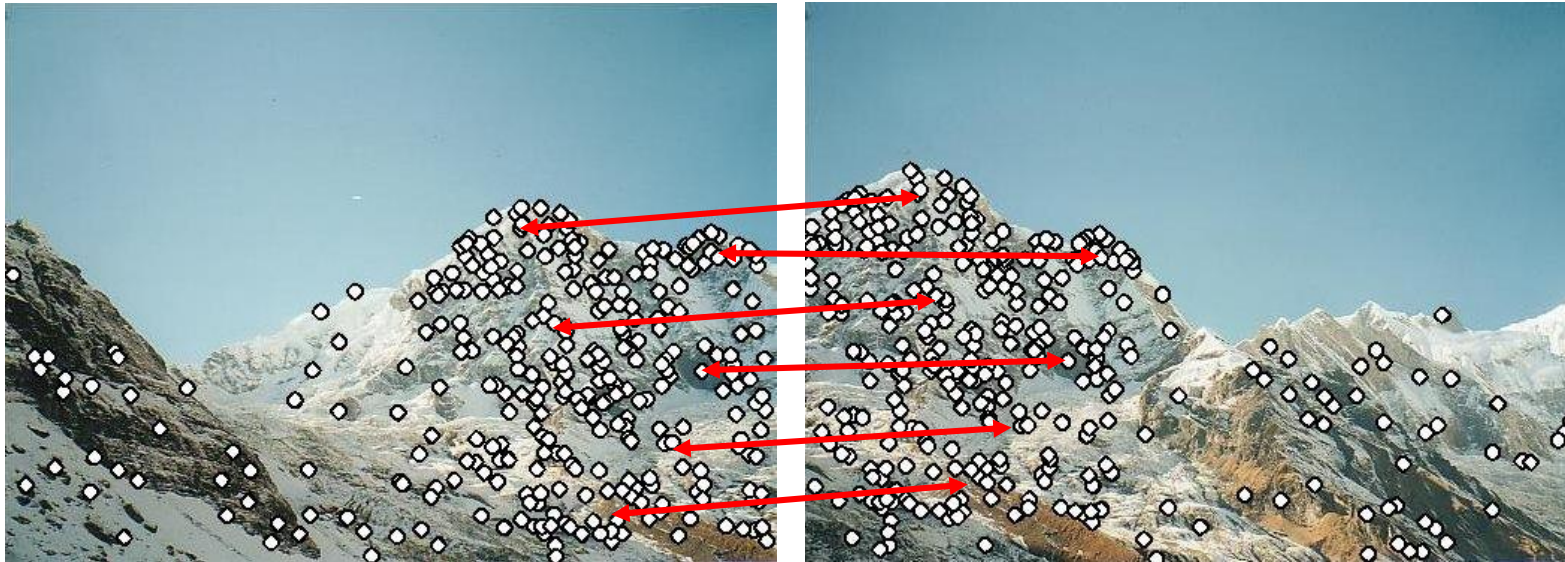
Why extract features?

- Motivation: panorama stitching
 - We have two images – how do we combine them?



Why extract features?

- Motivation: panorama stitching
 - We have two images – how do we combine them?



Step 1: extract features
Step 2: match features

Why extract features?

- Motivation: panorama stitching
 - We have two images – how do we combine them?

Step 1: extract features

Step 2: match features

Step 3: align images



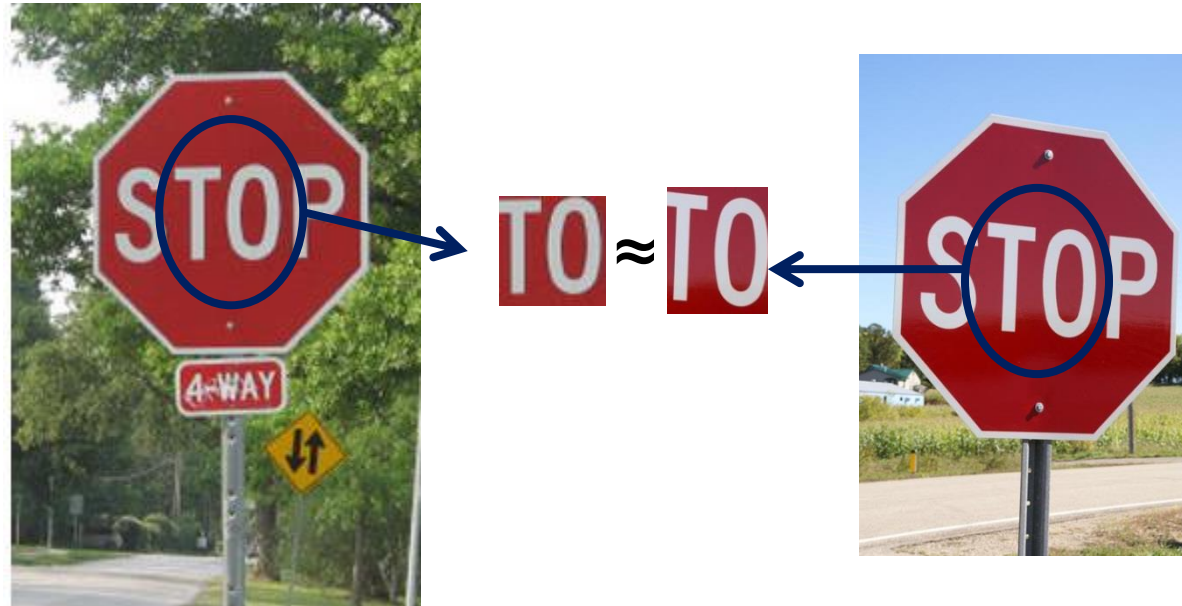


- Overview of correspondence and alignment
- Interesting points
- Harris detector
- Hessian detector

TODAY LECTURE

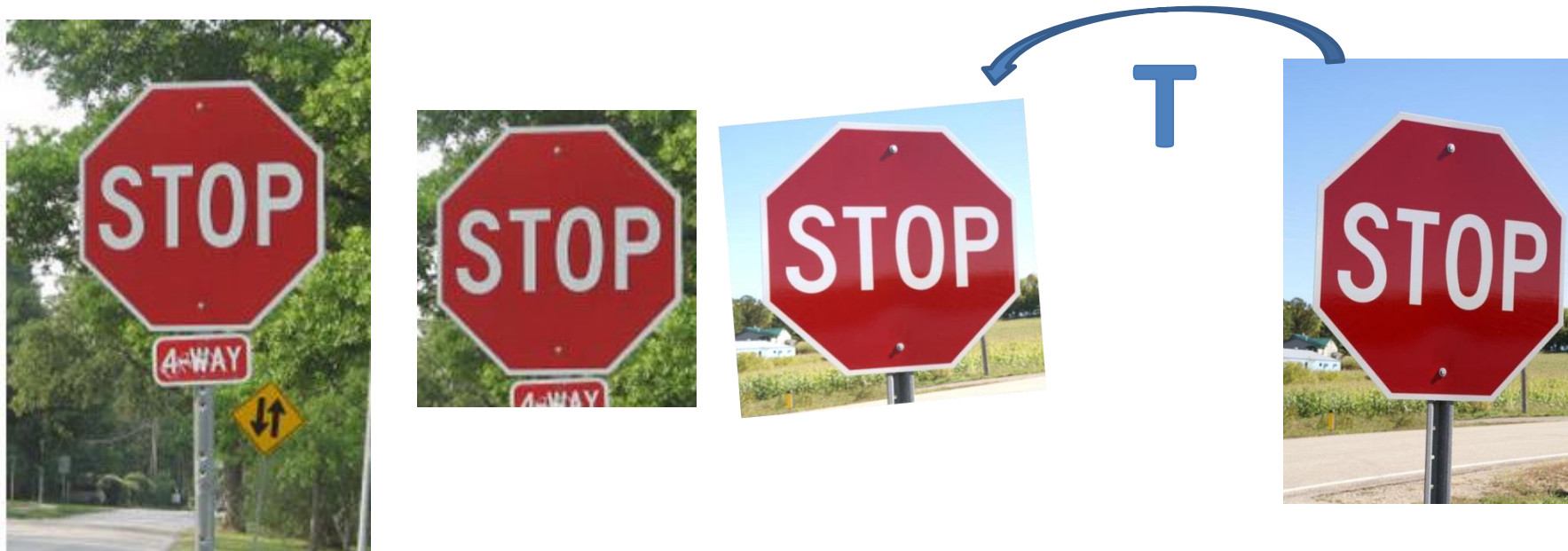
This section: correspondence and alignment

- Correspondence: matching points, patches, edges, or regions across images

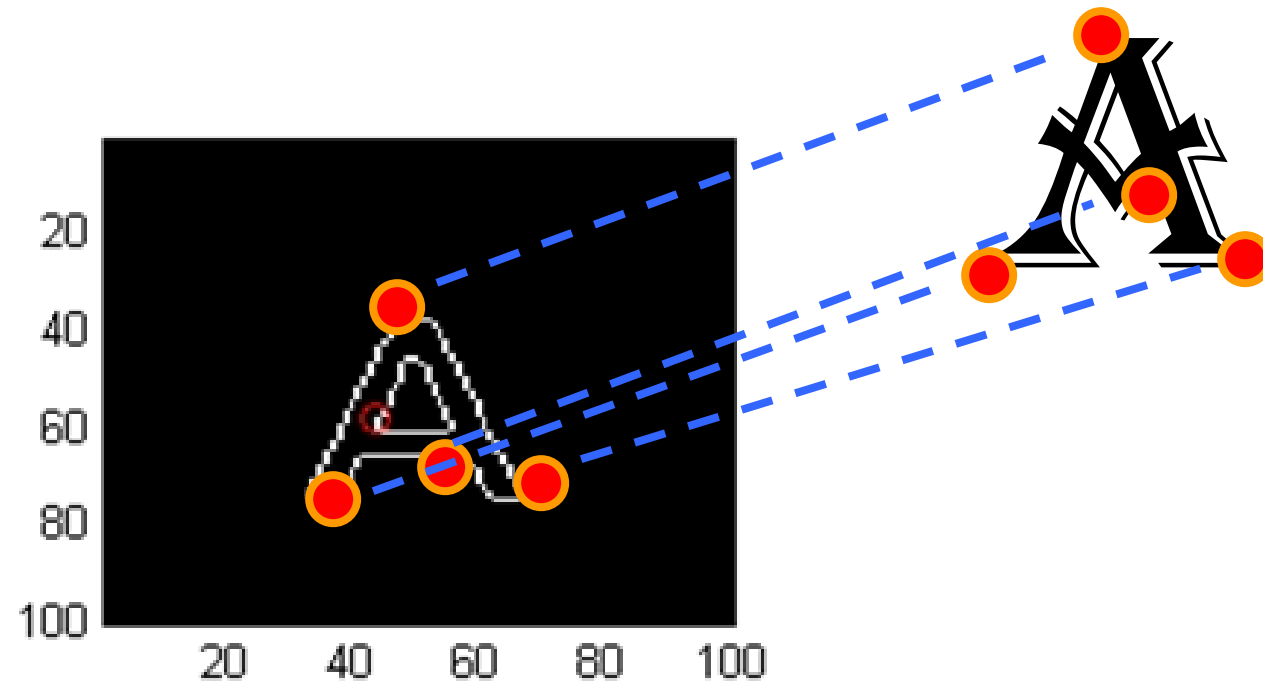


This section: correspondence and alignment

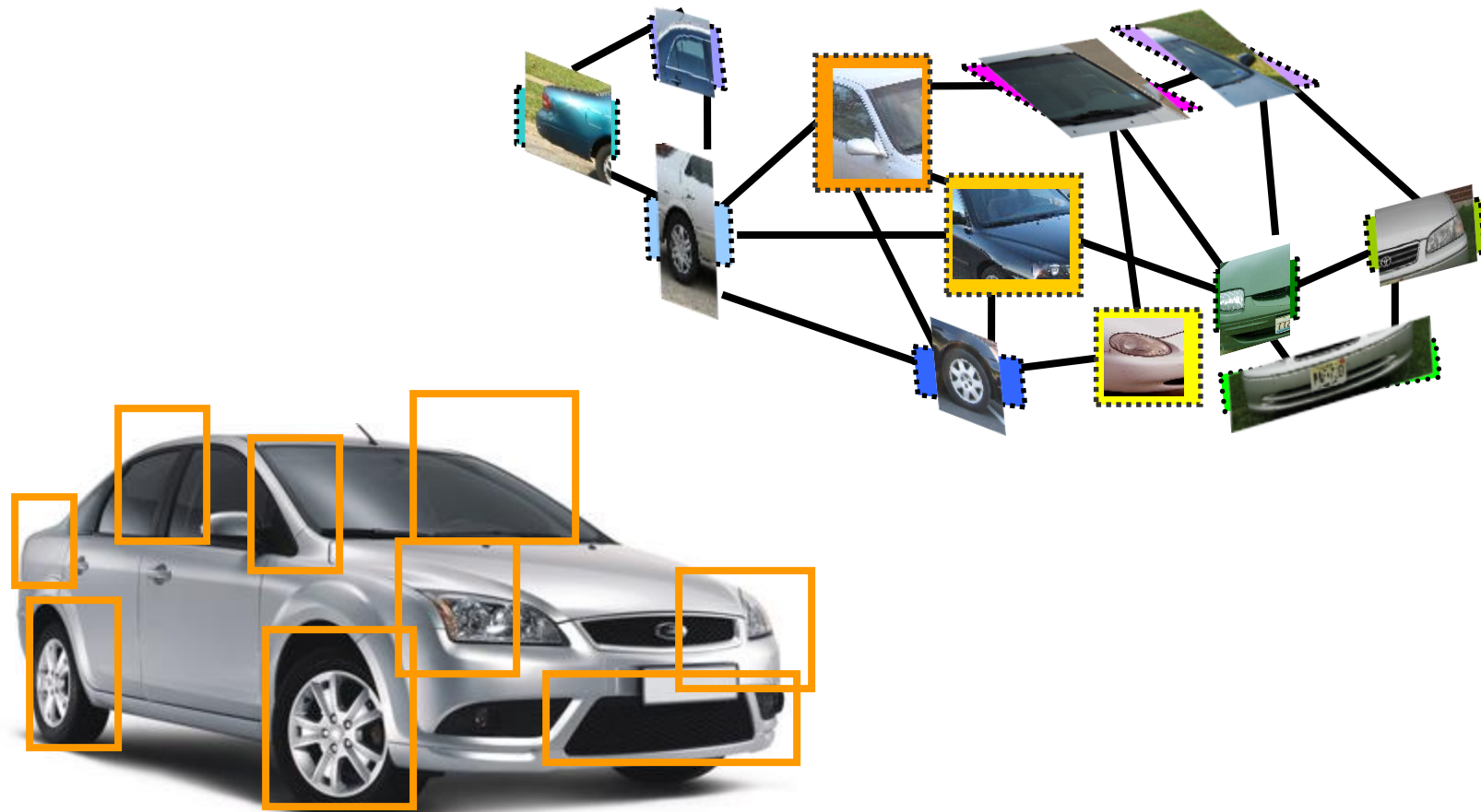
- Alignment: solving the transformation that makes two things match better



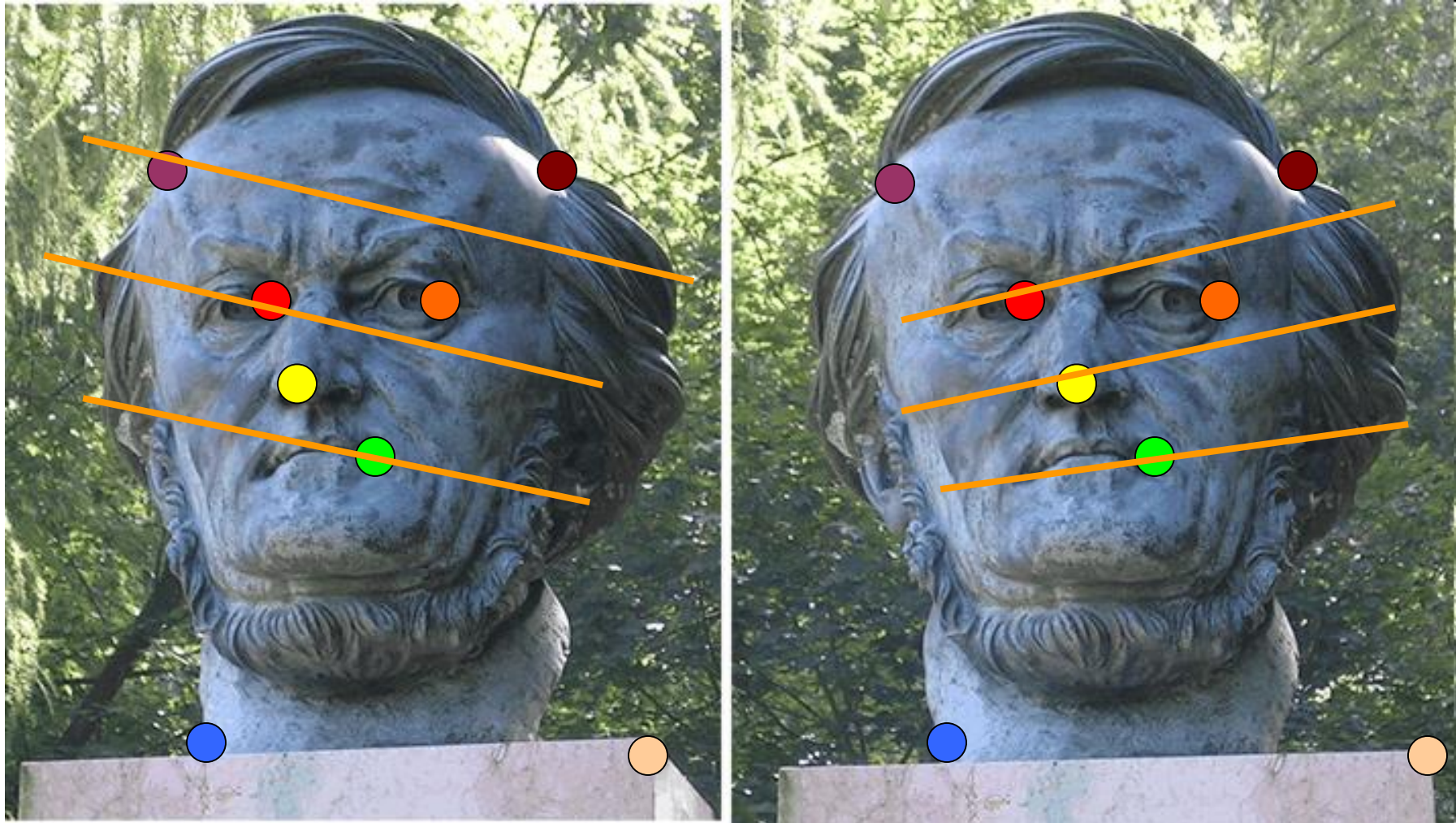
Example: fitting an 2D shape template



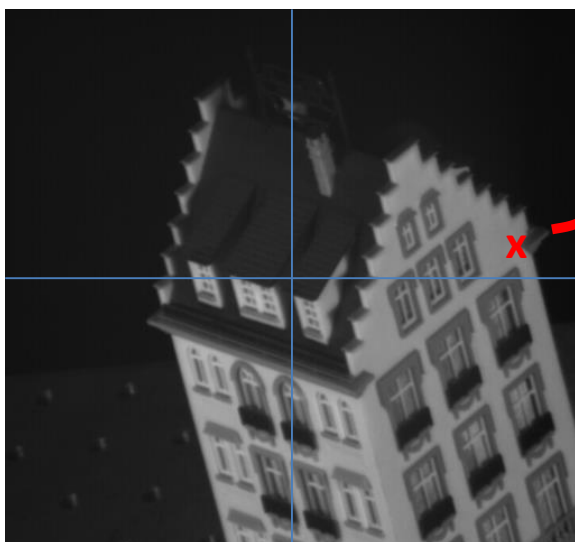
Example: fitting a 3D object model



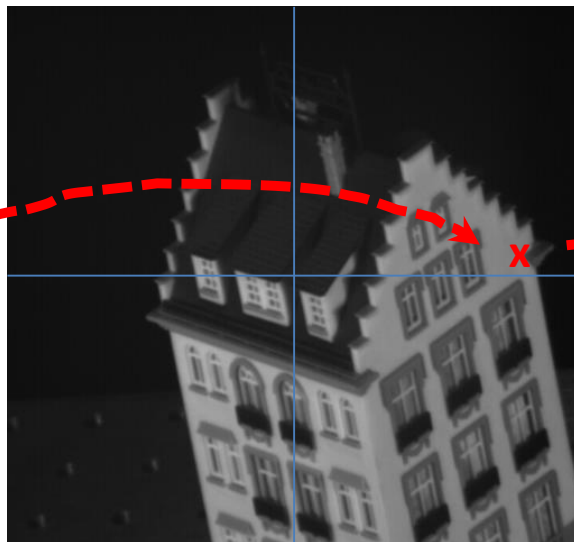
Example: estimating “fundamental matrix” that corresponds two views



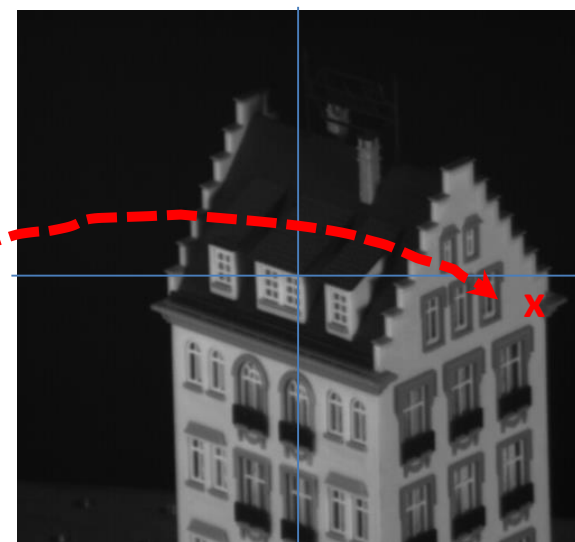
Example: tracking points



frame 0



frame 22

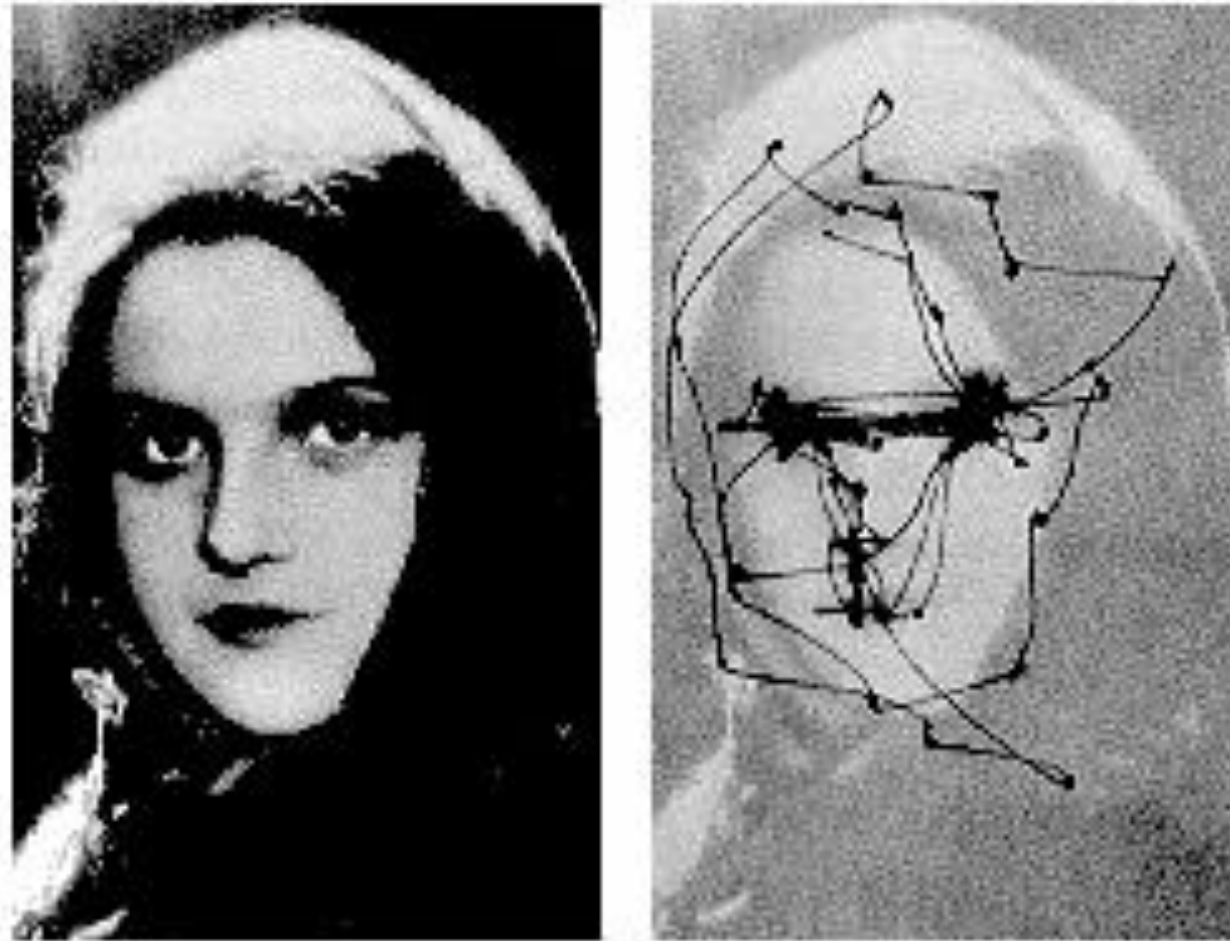


frame 49

Interest points

- Note: “interest points” = “keypoints”, also sometimes called “features”
- Many applications
 - tracking: which points are good to track?
 - recognition: find patches likely to tell us something about object category
 - 3D reconstruction: find correspondences across different views

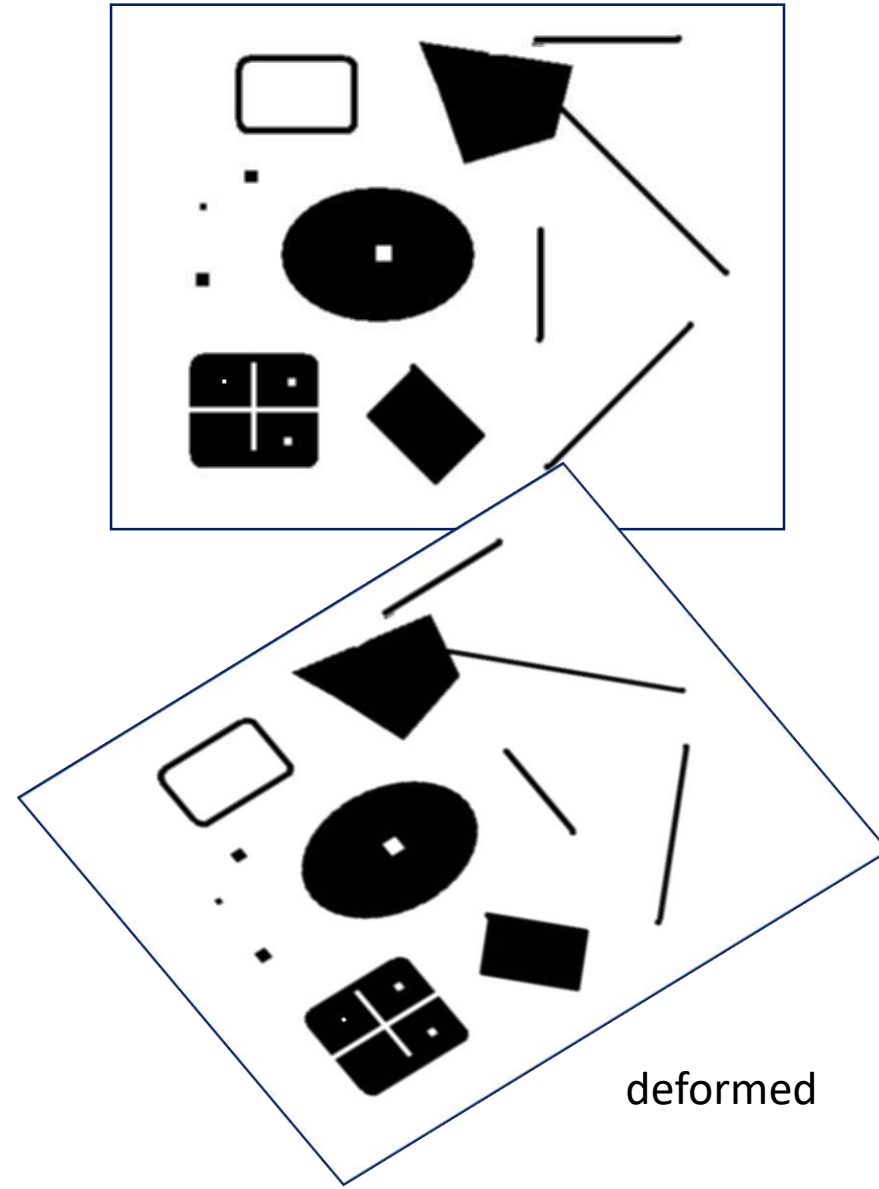
Human eye movements



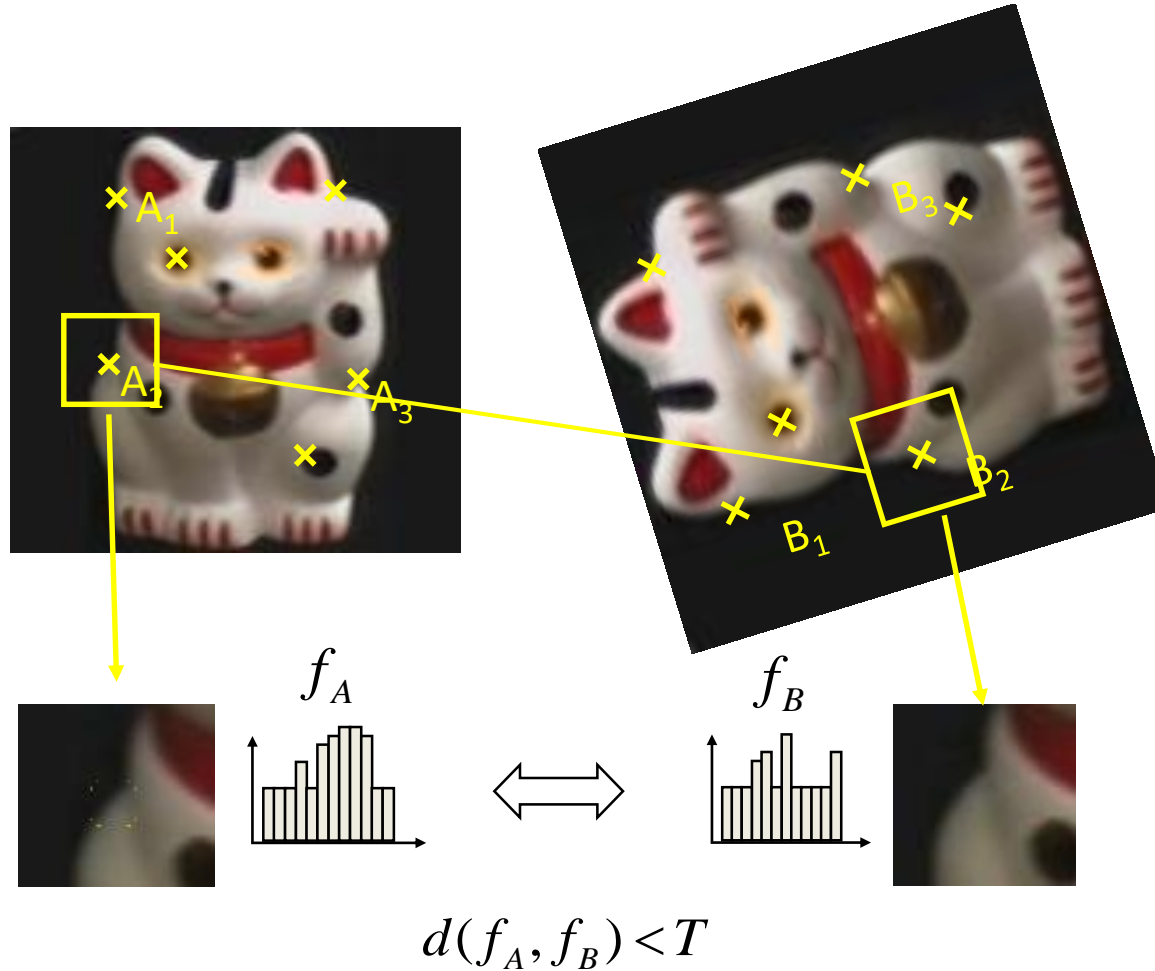
Yarbus eye tracking

This class: interest points

- Suppose you have to click on some point, go away and come back after I deform the image, and click on the same points again.
 - Which points would you choose?



Overview of Keypoint Matching



1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Goals for Keypoints



Detect points that are *repeatable* and *distinctive*

Choosing interest points

Where would you tell your friend to meet you?



Choosing interest points

Where would you tell your friend to meet you?



Many Existing Detectors Available

Hessian & Harris

Laplacian, DoG

Harris-/Hessian-Laplace

Harris-/Hessian-Affine

EBR and IBR

MSER

Salient Regions

Others...

[Beaudet '78], [Harris '88]

[Lindeberg '98], [Lowe 1999]

[Mikolajczyk & Schmid '01]

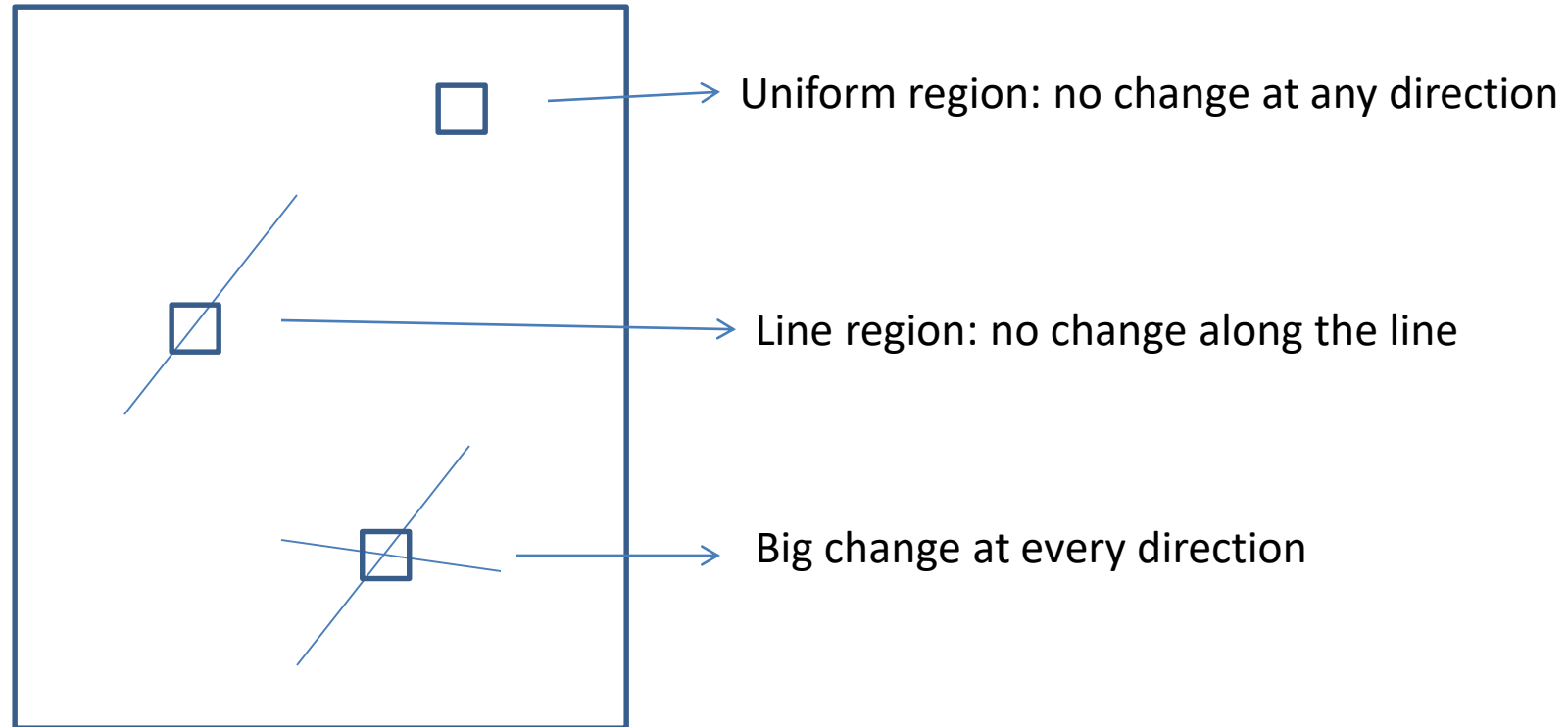
[Mikolajczyk & Schmid '04]

[Tuytelaars & Van Gool '04]

[Matas '02]

[Kadir & Brady '01]

Interesting points detection

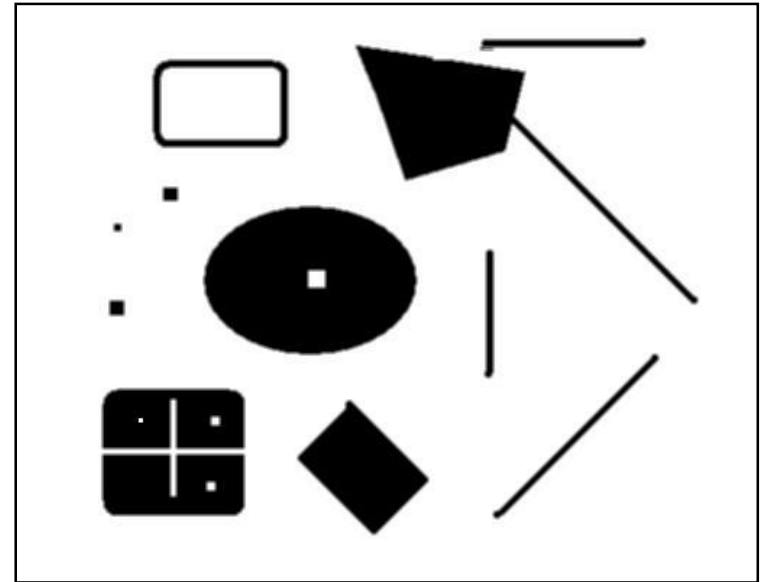


Harris Detector [Harris88]

- Second moment matrix

$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

Intuition: Search for local neighborhoods where the image content has two main directions (eigenvectors).

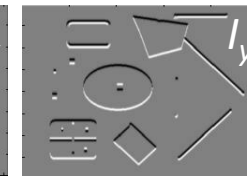
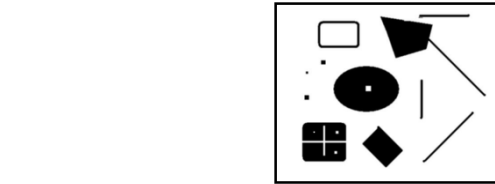


Harris Detector [Harris88]

- Second moment matrix

$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

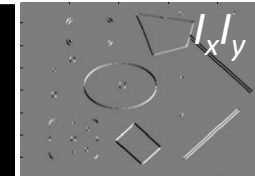
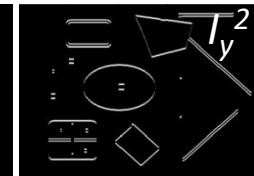
1. Image derivatives
(optionally, blur first)



$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$

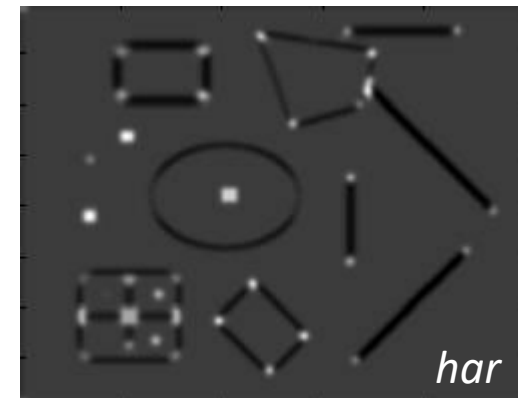


4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha [\text{trace}(\mu(\sigma_I, \sigma_D))]^2 =$$

$$g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha [g(I_x^2) + g(I_y^2)]^2$$

5. Non-maxima suppression



Harris Detector: Mathematics

$$M = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Want large eigenvalues, and small ratio $\frac{\lambda_1}{\lambda_2} < t$

2. We know

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

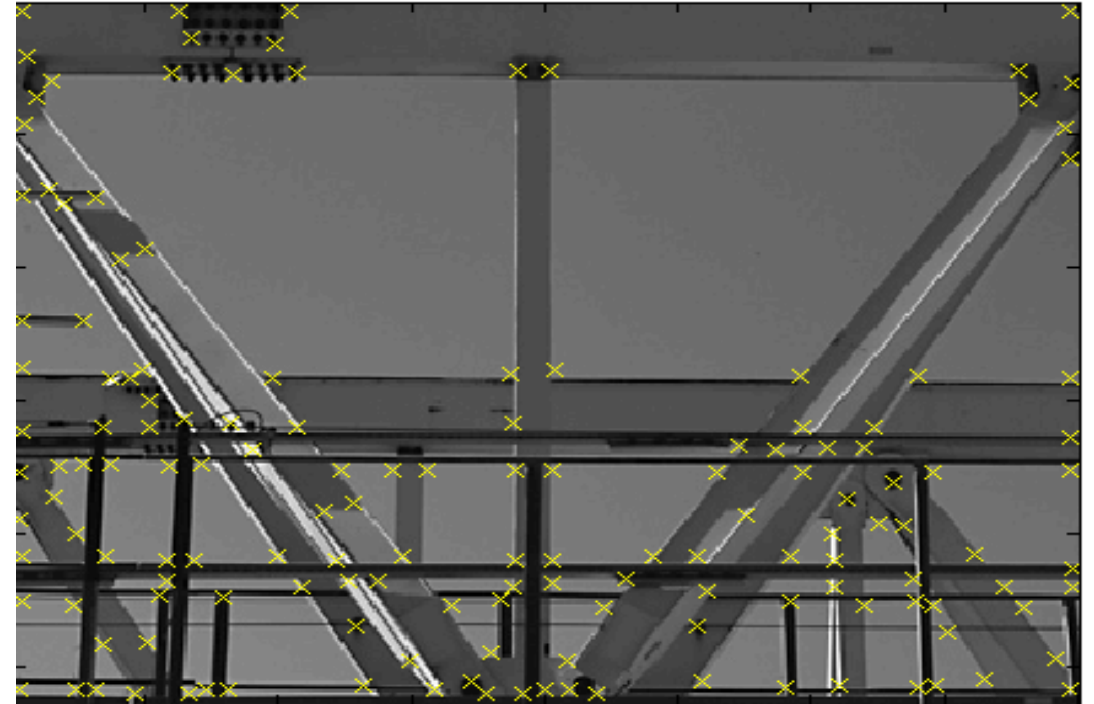
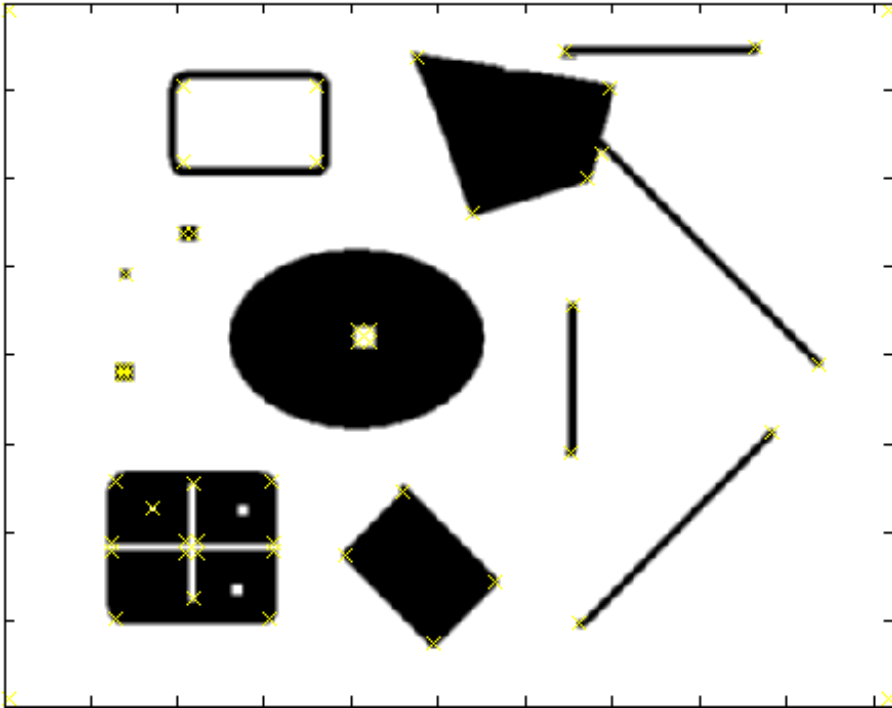
3. Leads to

$$\det M - k \cdot \text{trace}^2(M) > t$$

(k : empirical constant, $k = 0.04-0.06$)

Harris Detector – Responses

[Harris88]



Effect: A very precise corner detector.

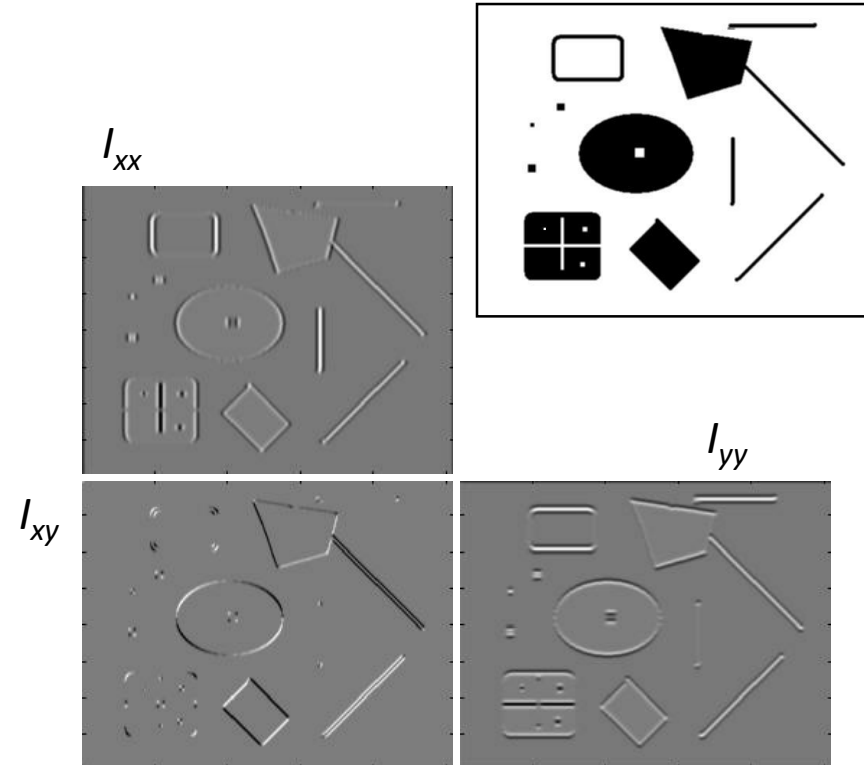
Harris Detector - Responses [Harris88]



Hessian Detector [Beaudet78]

- Hessian determinant

$$\text{Hessian}(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$



Intuition: Search for strong curvature in two orthogonal directions

Hessian Detector [Beaudet78]

- Hessian determinant

$$\text{Hessian}(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

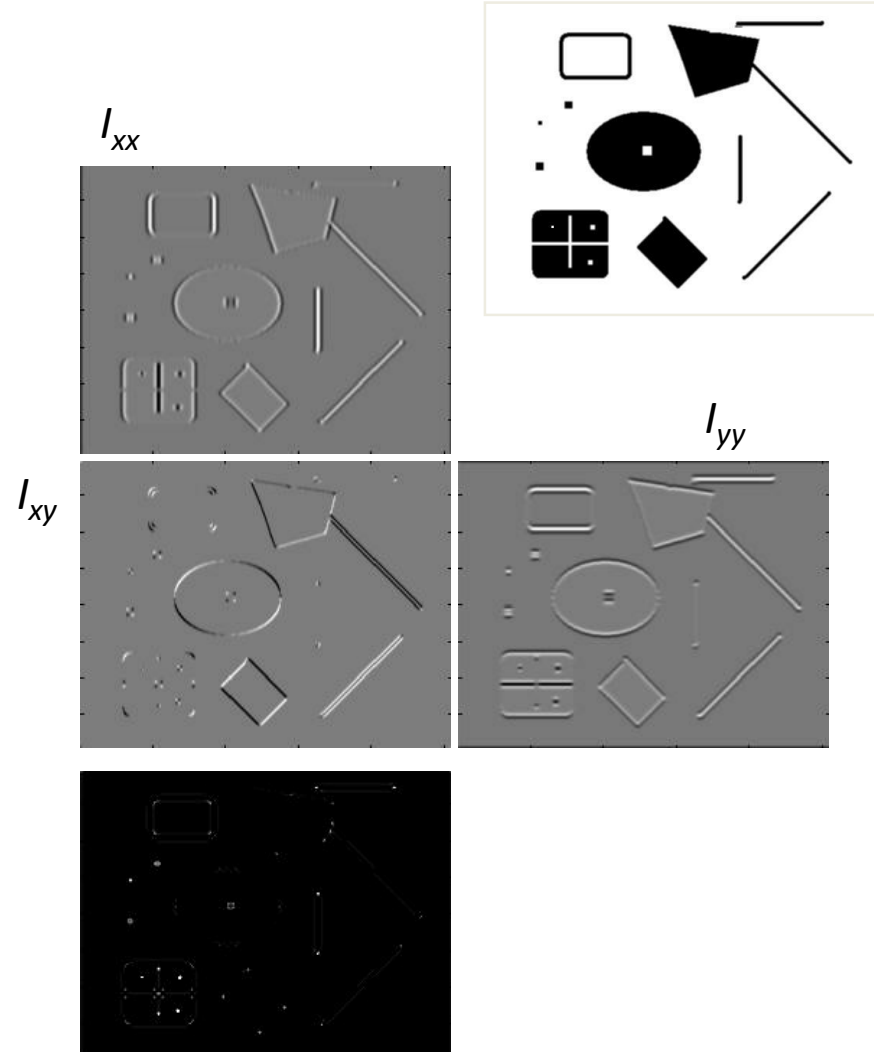
$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

$$\det(\text{Hessian}(I)) = I_{xx} I_{yy} - I_{xy}^2$$

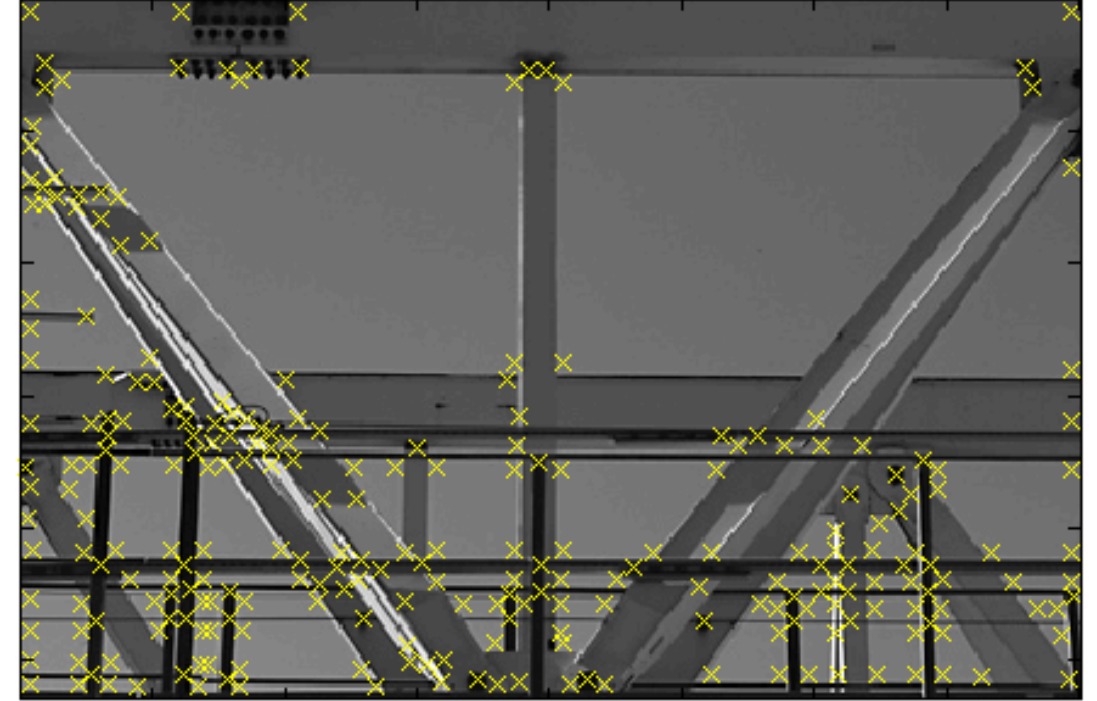
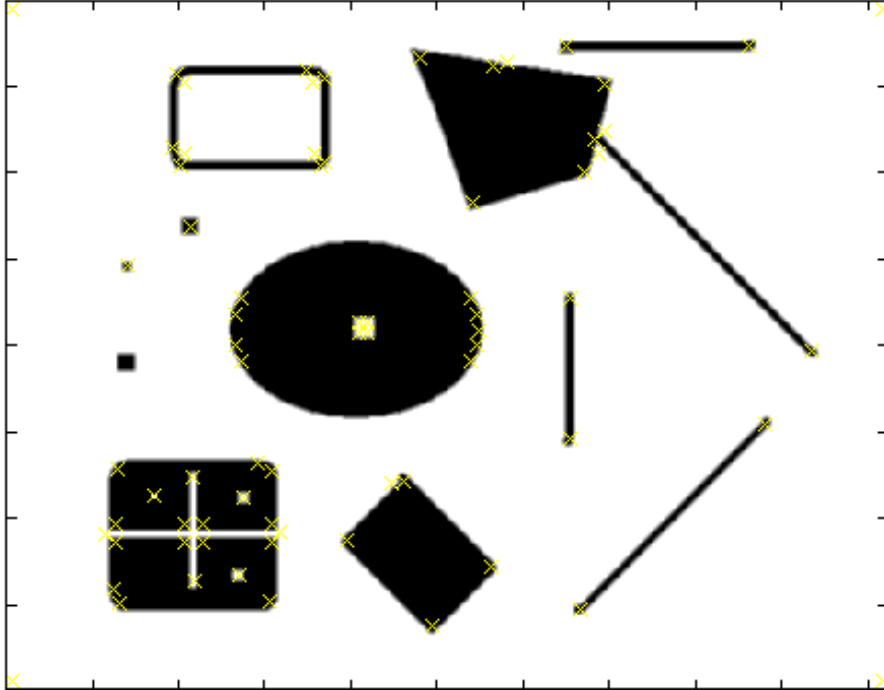
In Matlab:

$$I_{xx} \cdot I_{yy} - (I_{xy})^2$$



Hessian Detector – Responses

[Beaudet78]



Effect: Responses mainly on corners and strongly textured areas.

Hessian Detector – Responses [Beaudet78]



So far: can localize in x-y, but not scale

