

Feature Detection and Matching

Chul Min Yeum

Assistant Professor

Civil and Environmental Engineering

University of Waterloo, Canada

CIVE 497 – CIVE 700: Smart Structure Technology



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

Last updated: 2020-02-26

Two Major Challenges in Computer Vision

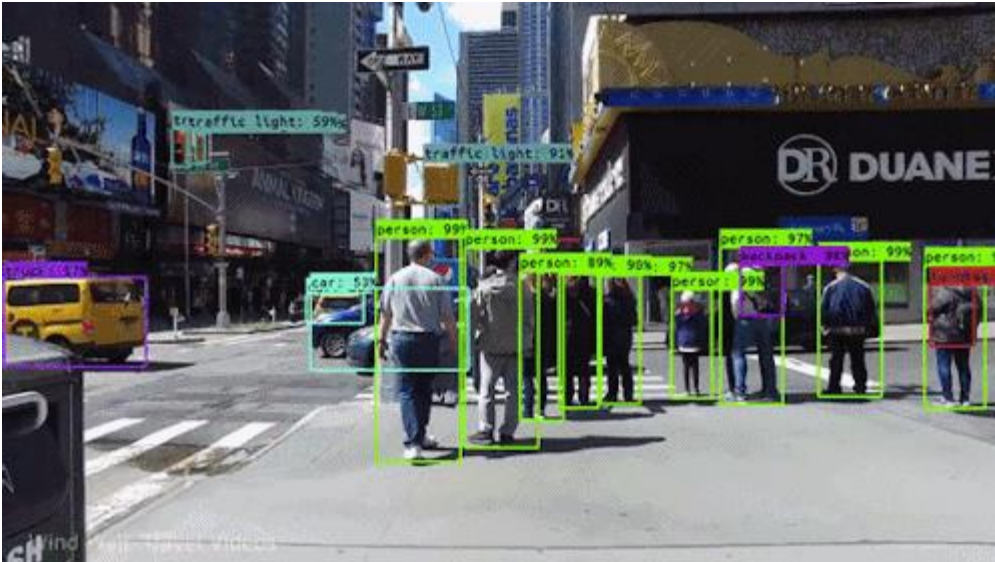
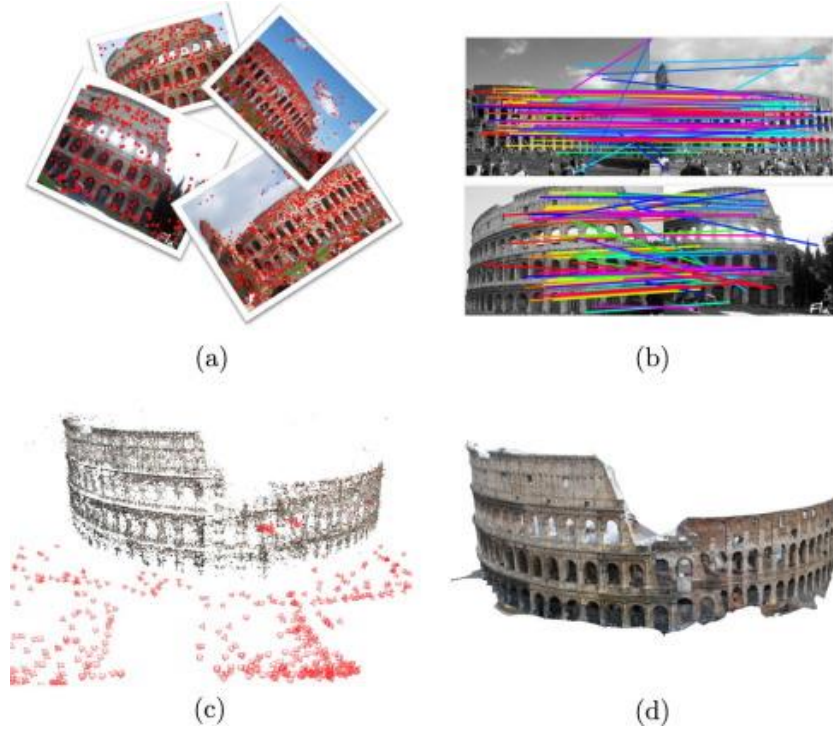


Image recognition



Feature matching

Why We Extract Features? Extracting Features

Motivation: panorama stitching

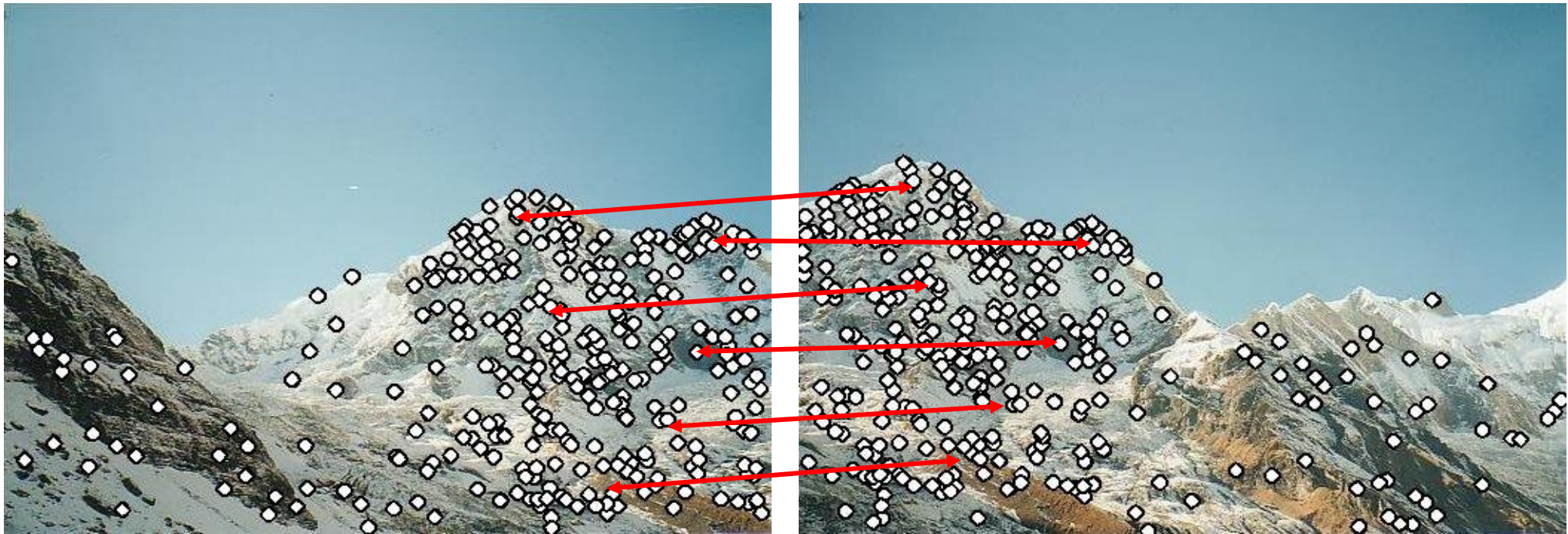
- We have two images
- How do we combine them?



Why We Extract Features? Extracting Features (Continue)

Motivation: panorama stitching

- We have two images
- How do we combine them?



Why We Extract Features? Extracting Features (Continue)

Motivation: panorama stitching

- We have two images
- How do we combine them?



Example: Automatic Panoramas

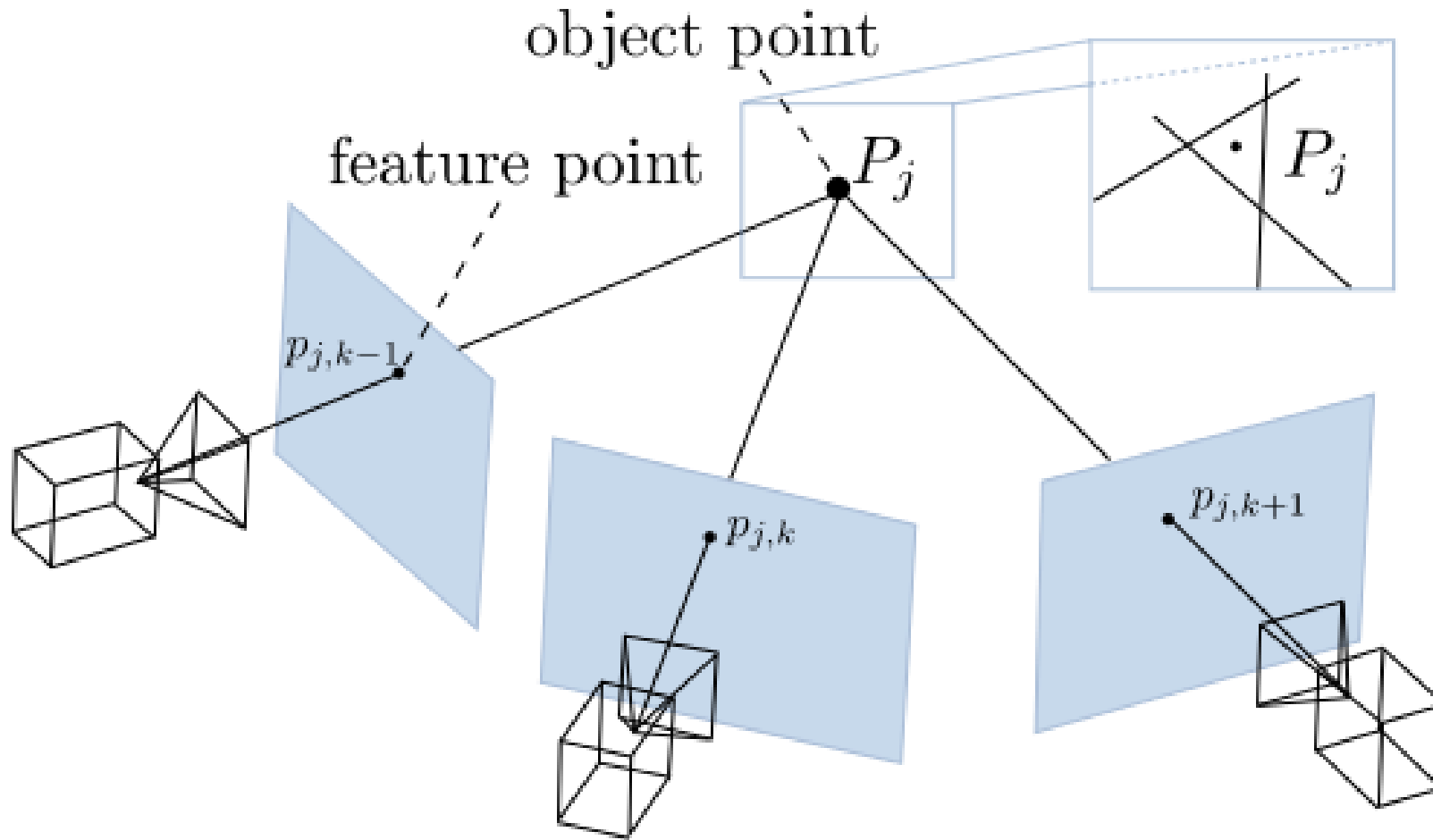


A large volume collected images from drone



Orthophoto($10,000 \times 3,656$) geometrically connected to each collected images

Example: Multi-view Geometry



Stereo-camera

Image Matching



What are Features (Keypoints)?

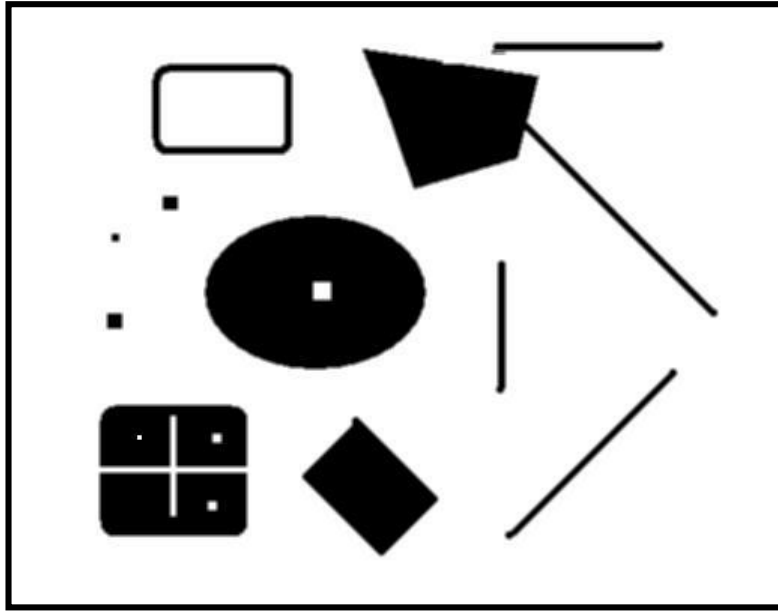
- A feature is an prominent point that is selected based on a certain criteria, such as edge, corner, or blob.
- This is represented in terms of the coordinates of the image points by pixel or sub-pixel.
- The feature likely contain and preserve the distinctive local regional information.
- Note: “interest points” = “keypoints”, also sometimes called “features”

Many applications:

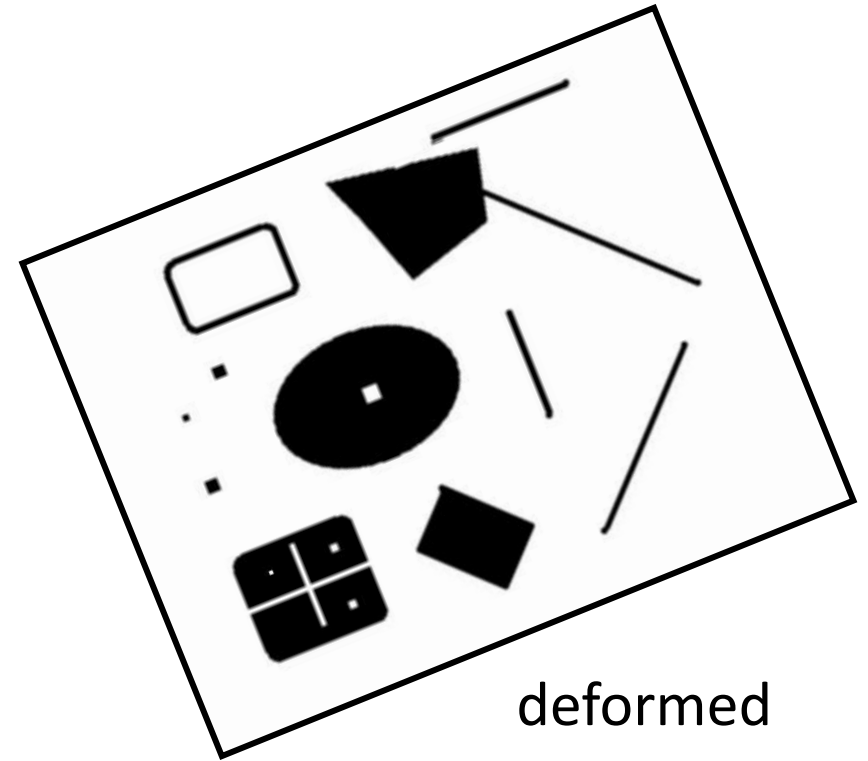
- Object/motion tracking: which points are good to track?
- Object recognition: find patches likely to tell us something about object category
- 3D scene reconstruction: find correspondences across different views

Example: Keypoints/Features

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?

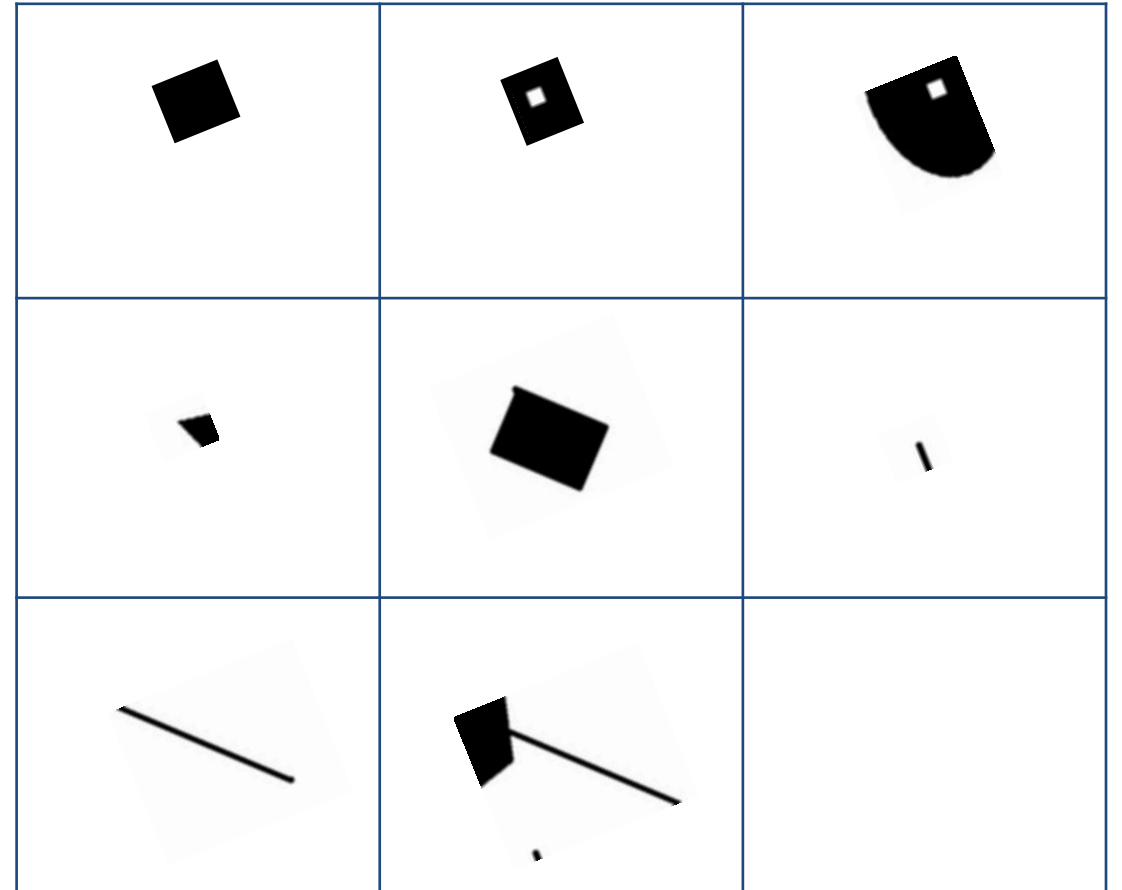
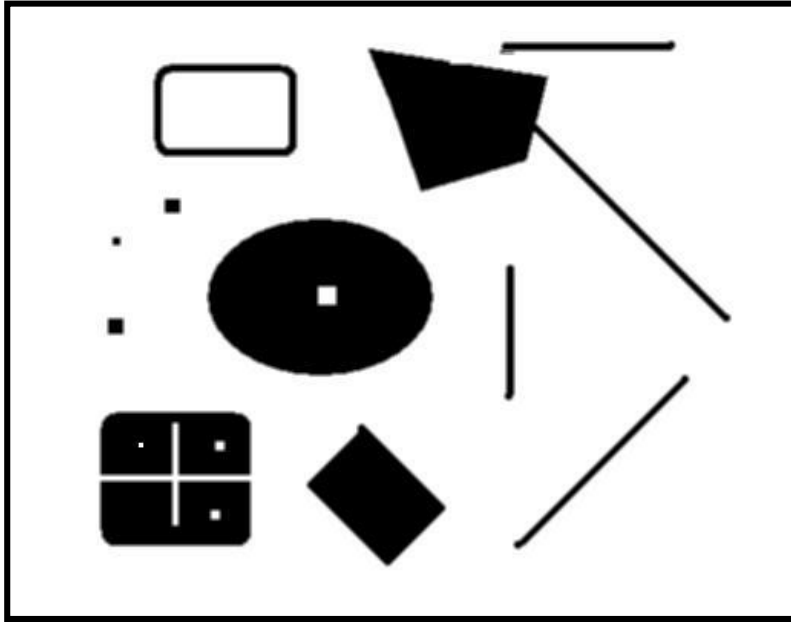


original



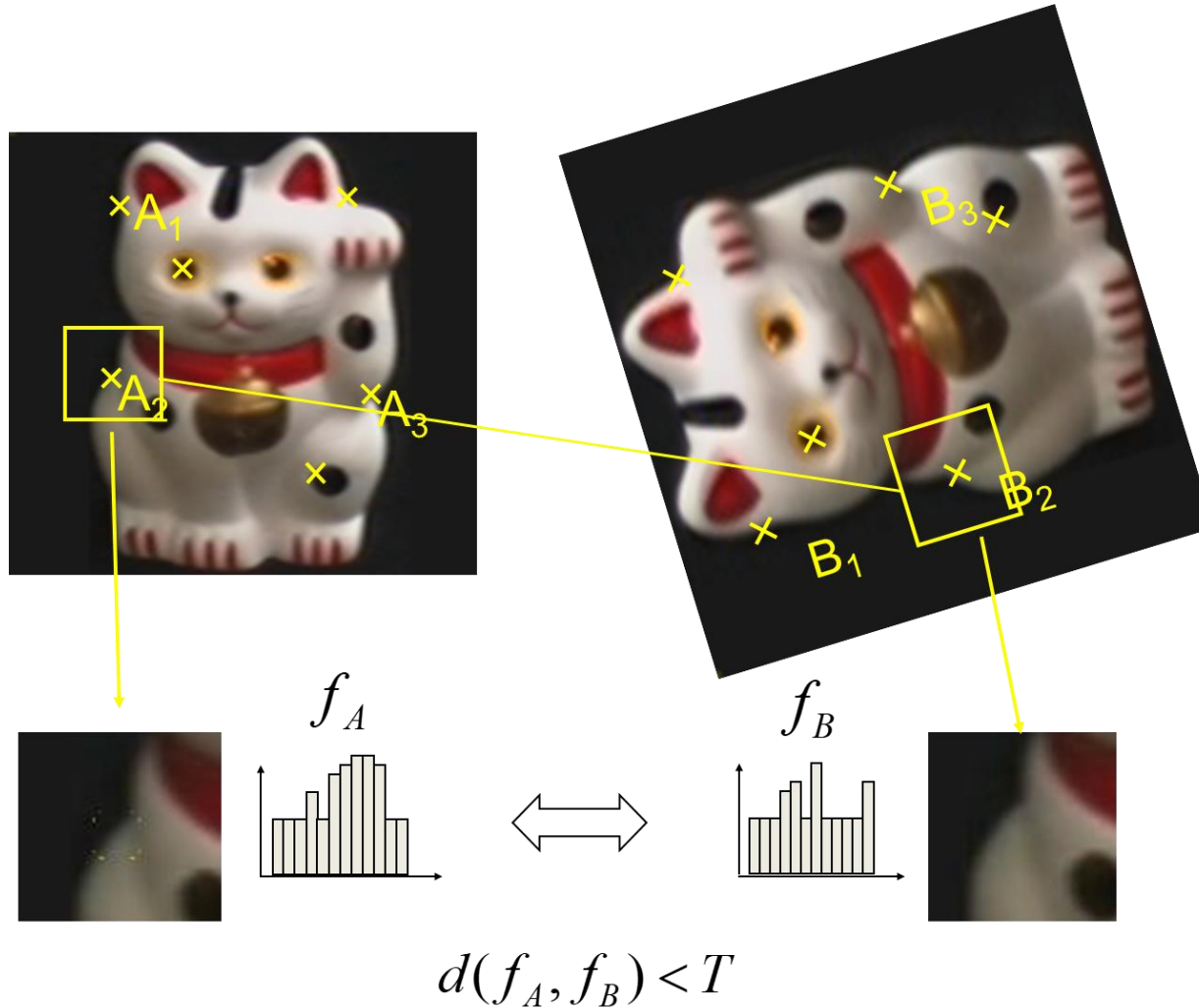
deformed

Example: Keypoints/Features (Continue)



- A feature is an prominent point that is selected based on a certain criteria, such as edge, corner, or blob.
- The feature likely contain and preserve the distinctive local regional information.

Overview of Feature Matching



1. Find a set of distinctive keypoints
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

Local?

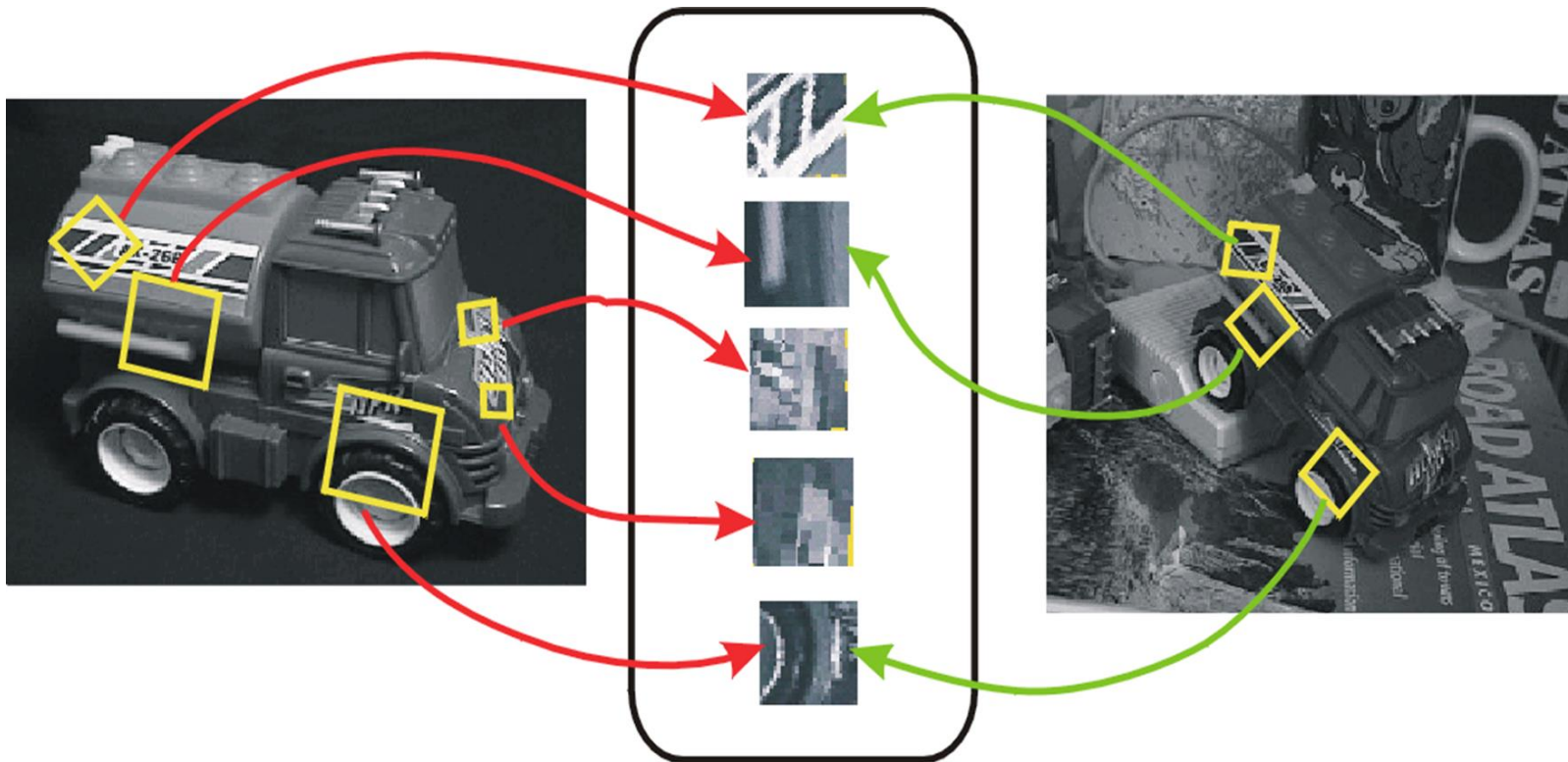
BEST Features !!



Detect points that are *repeatable* and *distinctive*

Invariant Local Features

Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



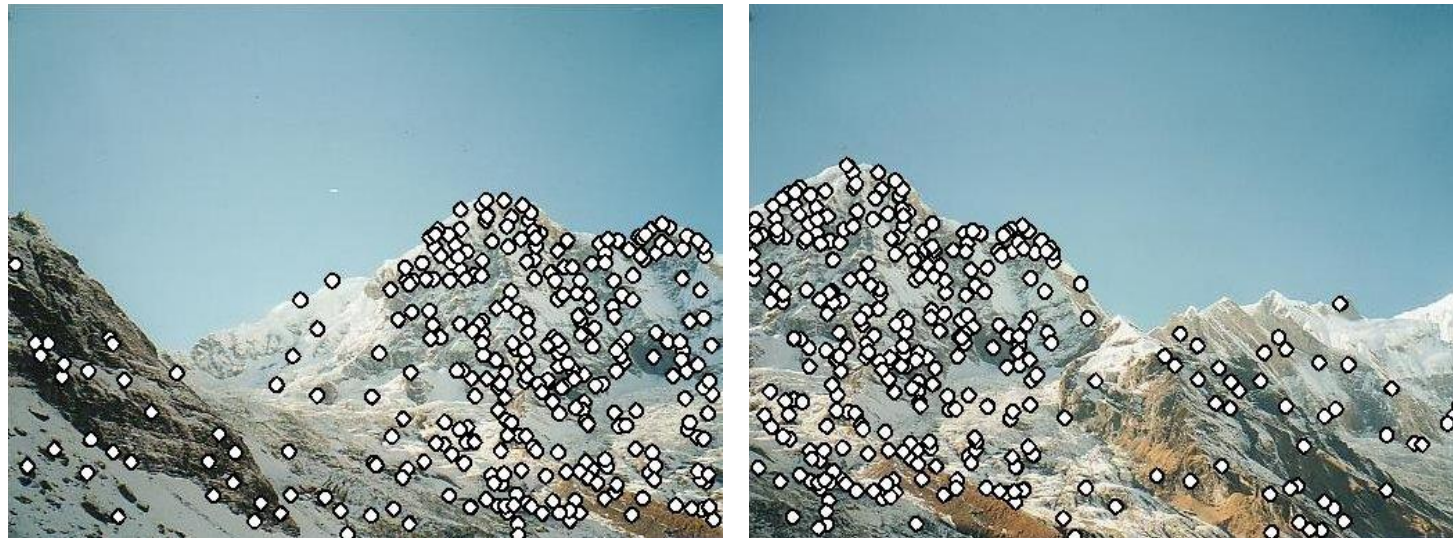
Characteristic of Good Features

Repeatability: The same feature can be found in several images despite geometric and photometric transformations

Saliency: Each feature is distinctive

Compactness and efficiency: Many fewer features than image pixels

Locality: A feature occupies a relatively small area of the image; robust to clutter and occlusion

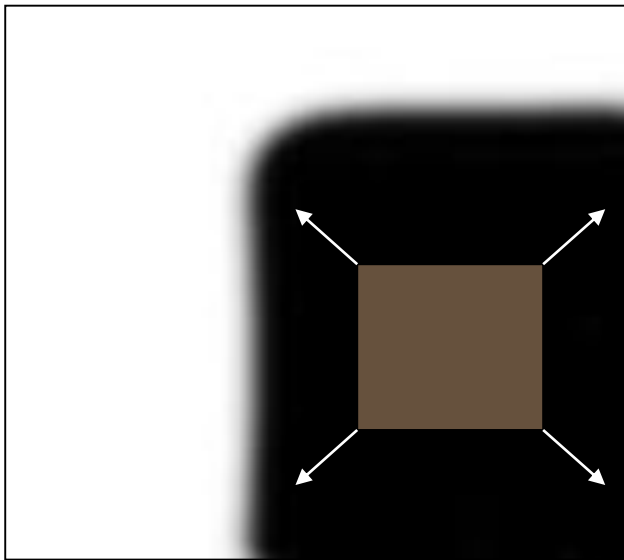


What Points would You Choose?



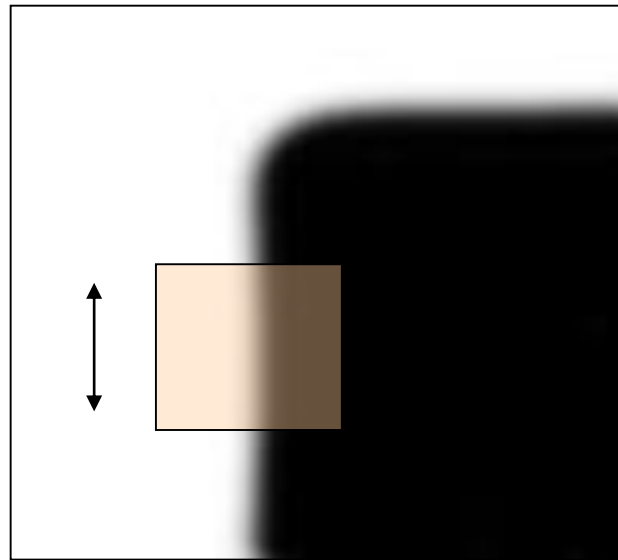
Corner Detection: Basic Idea

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give a *large change* in intensity



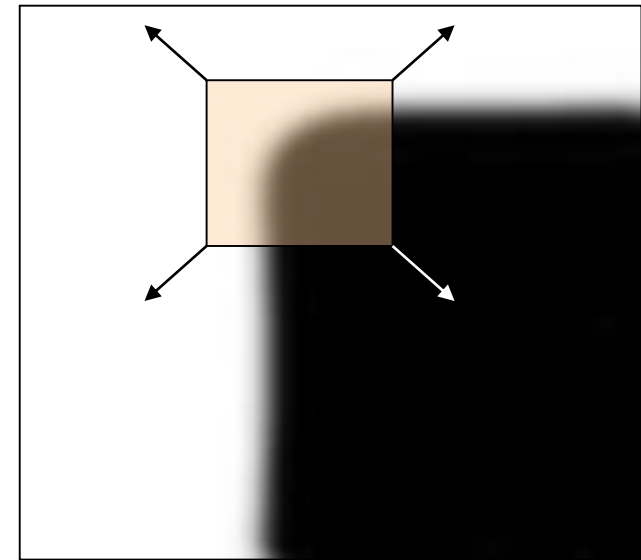
Flat region

no change in all
directions



Edge

no change along the edge
direction



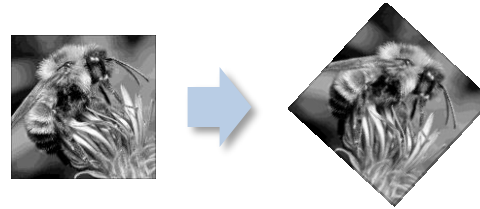
Corner

significant change in all
directions

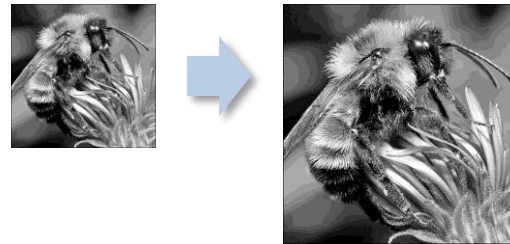
Image Transformations

- Geometric

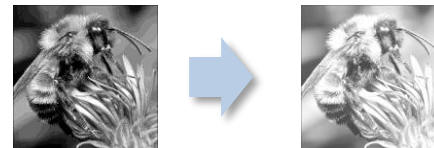
Rotation



Scale



- Photometric
Intensity change



Invariance and Covariance

We want corner locations to be *invariant* to photometric transformations and *covariant* to geometric transformations

Invariance: images are transformed and corner locations do not change

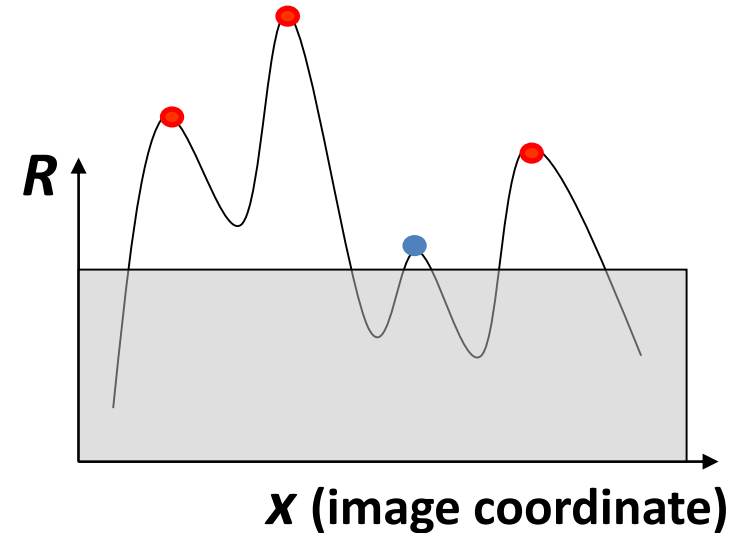
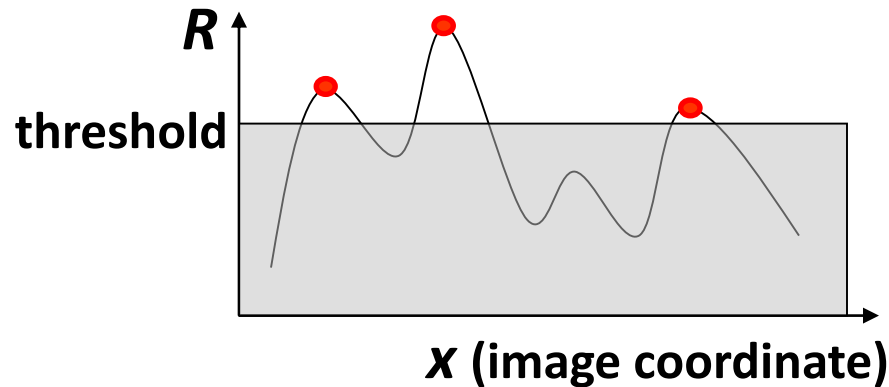
Covariance: if we have two transformed versions of the same image, features should be detected in corresponding locations

Intensity Change



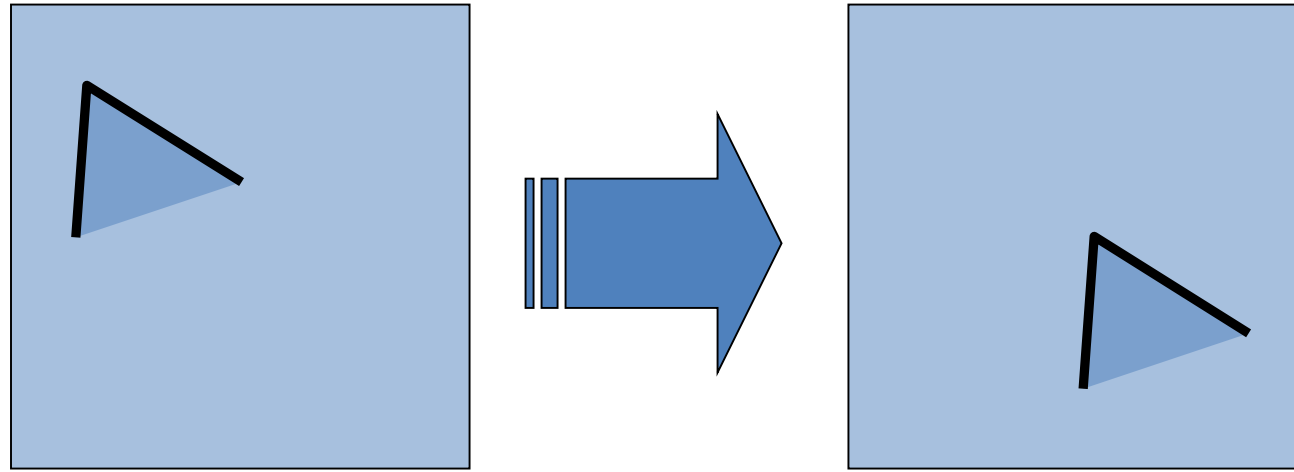
$$I \rightarrow a I + b$$

- Only derivatives are used \Rightarrow invariance to intensity shift $I \rightarrow I + b$
- Intensity scaling: $I \rightarrow a I$



Partially invariant to intensity change

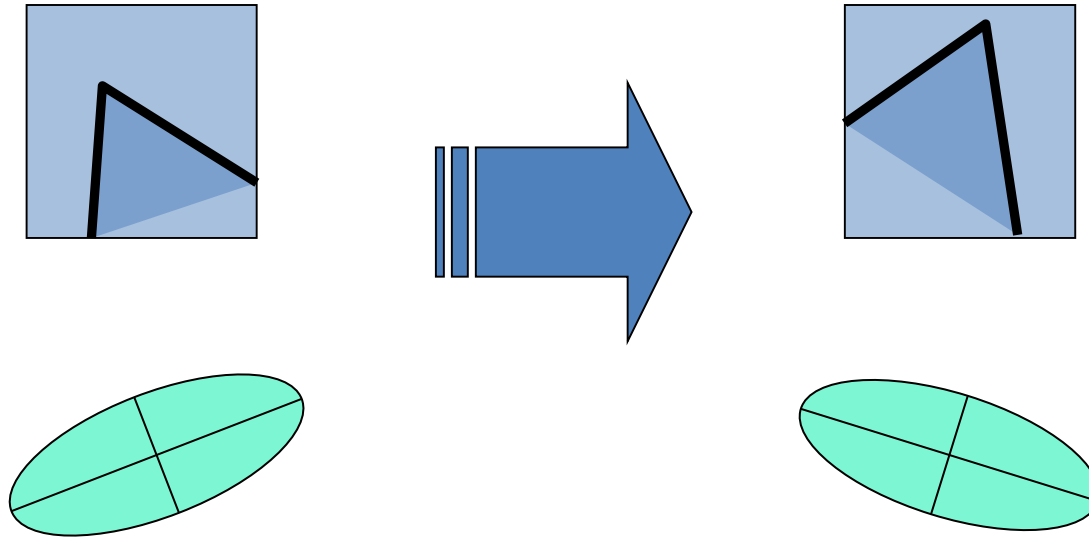
Image Translation



- Derivatives and window function are shift-invariant

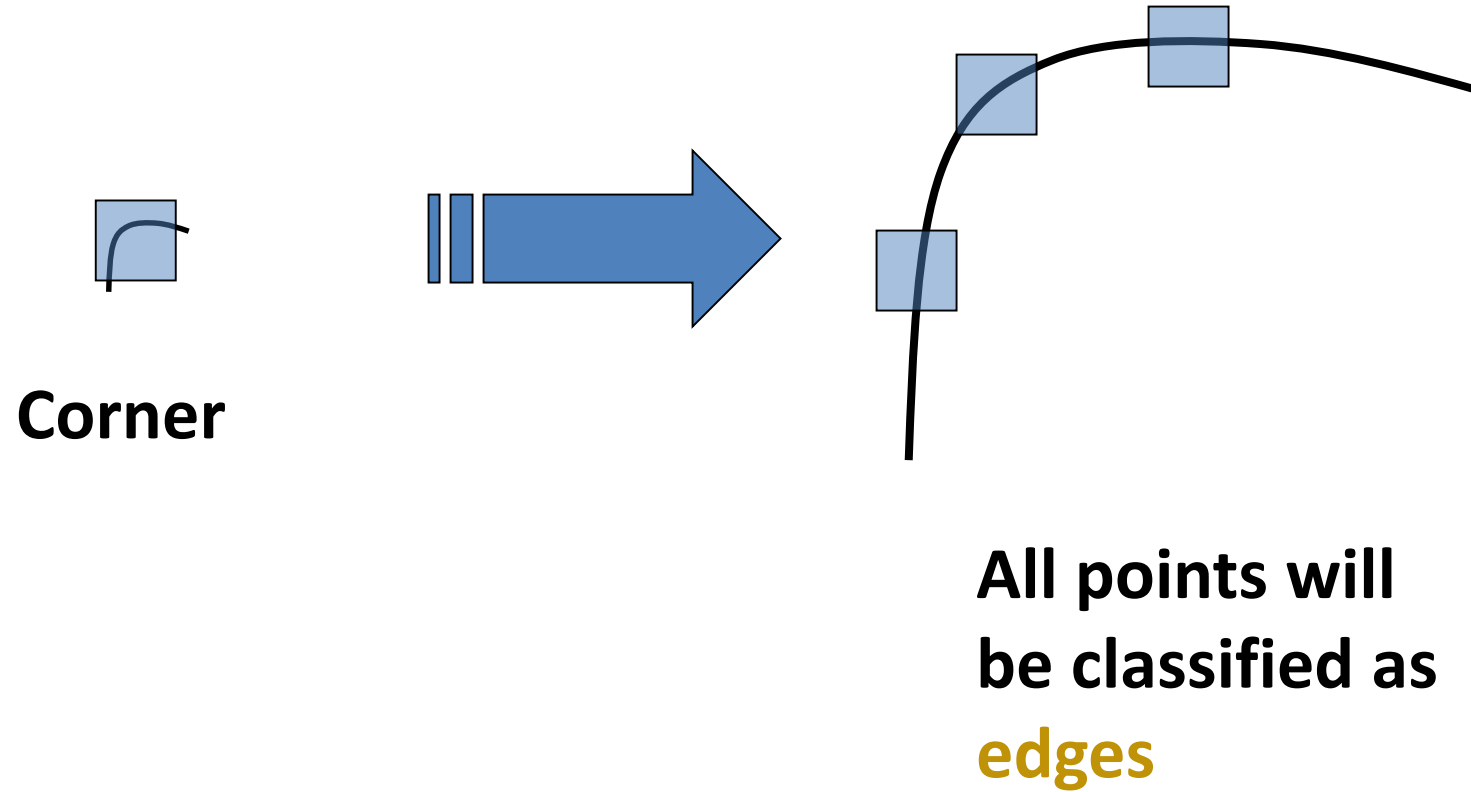
Corner location is covariant to image translation

Image Rotation



Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

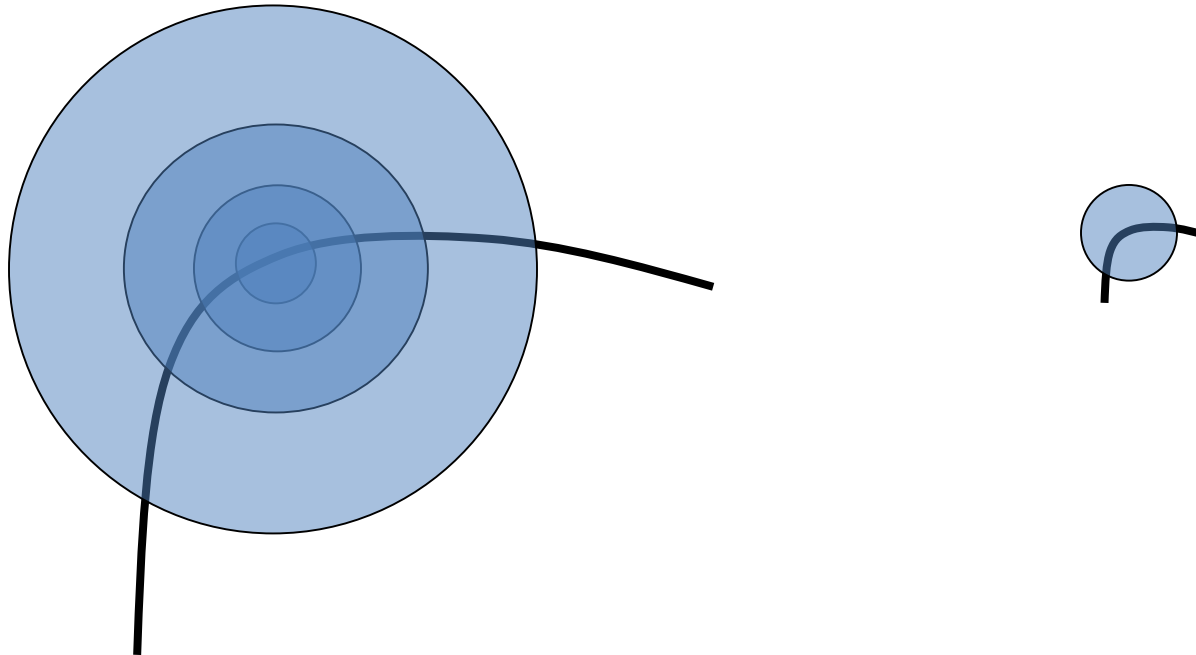
Corner location is covariant to image rotation



Corner location is **not** invariant to image scale!

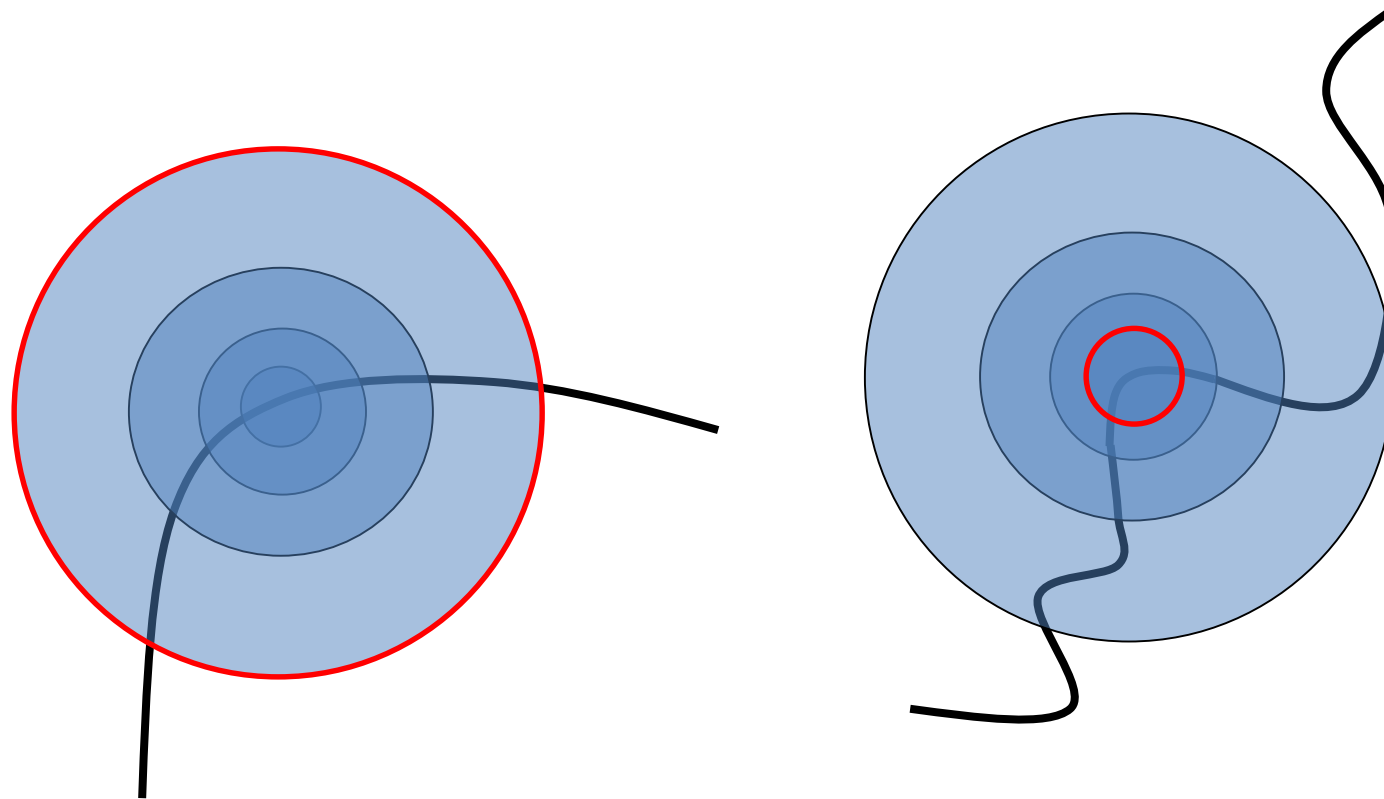
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



Scale Invariant Detection (Continue)

- The problem: how do we choose corresponding circles *independently* in each image?
- Choose the scale of the “best” corner



Example: Scale Invariance



Slide Credits and References

- Lecture notes: S. Narasimhan
- Lecture notes: Gordon Wetzstein
- Lecture notes: Mohammad Jahanshahi
- Lecture notes: Noah Snavely
- Lecture notes: L. Fei-Fei
- Lecture notes: D. Forsyth
- Lecture notes: James Hayes
- Lecture notes: Yacov Hel-Or
- Lecture notes: K. Grauman, B. Leibe