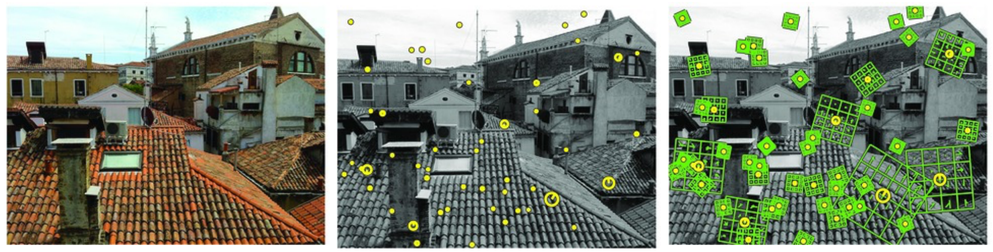
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 12: Correspondence and SIFT

Menu for Today

Topics:

- **Correspondence** Problem
- **Patch** matching
- **Invariance**, geometric, photometric
- **SIFT** = Scale Invariant Feature Transform

Readings:

- **Today's** Lecture: Szeliski Chapter 7, Forsyth & Ponce 5.4

Reminders:

- **Assignment 4**: RANSAC and Panorama Stitching — **now available**

Correspondence Problem

- A basic problem in Computer Vision is to establish matches (correspondences) between images
- This has **many** applications: rigid/non-rigid tracking, object recognition, image registration, structure from motion, stereo...

Image **Matching**

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Image **Matching**

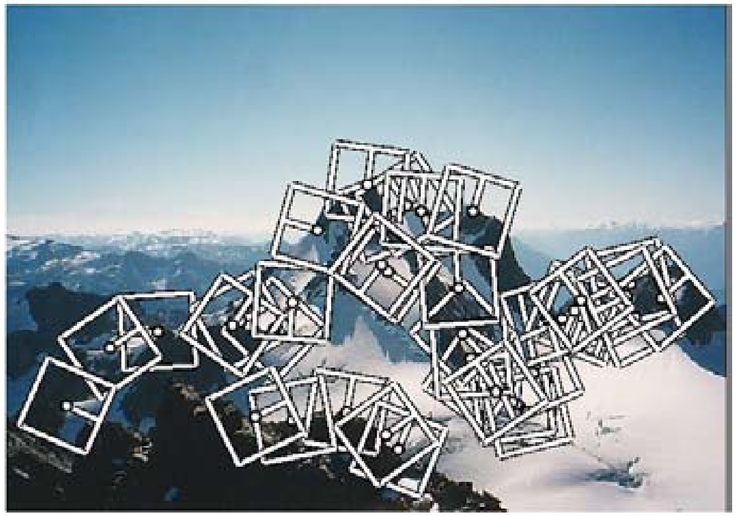
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Features and Matching

- Feature detectors
  - Canny edges, Harris corners, DoG, MSERs
- Feature descriptors
  - Image patches, invariance, SIFT, learned features

[Szeliski Chapter 7]

Feature Detectors



Corners/Blobs

Regions

Edges

Straight Lines

Feature Descriptors

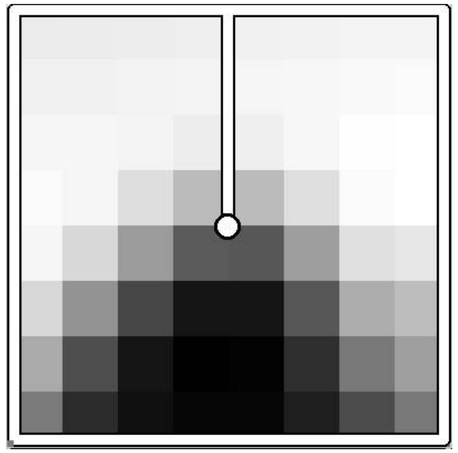
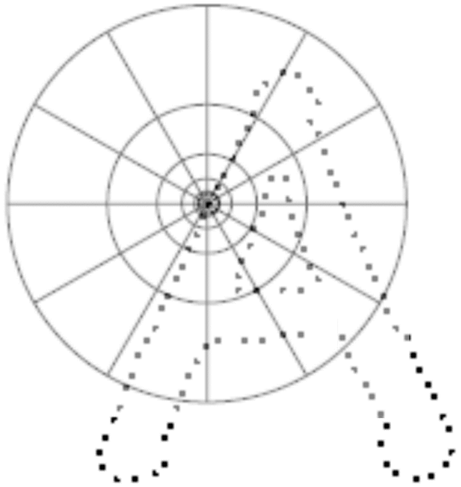
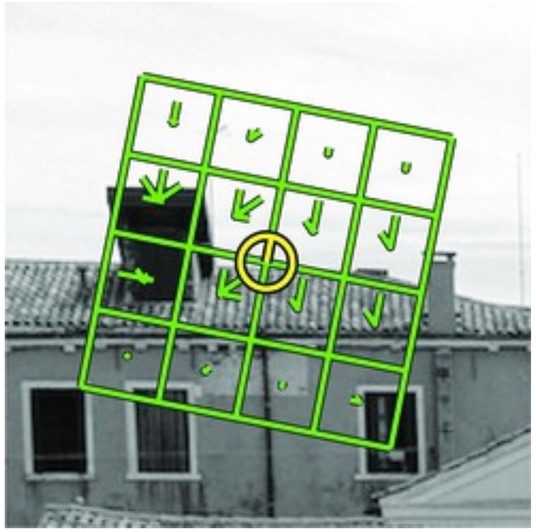


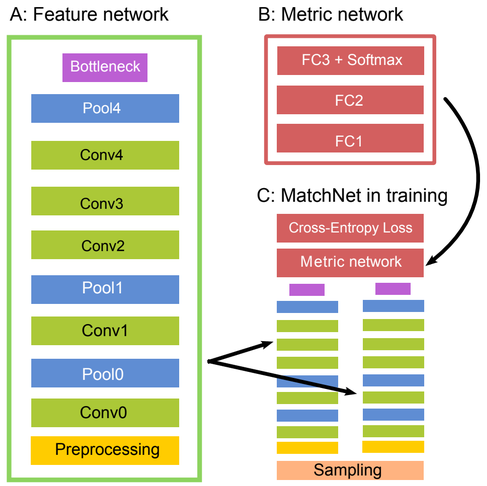
Image Patch



Shape Context



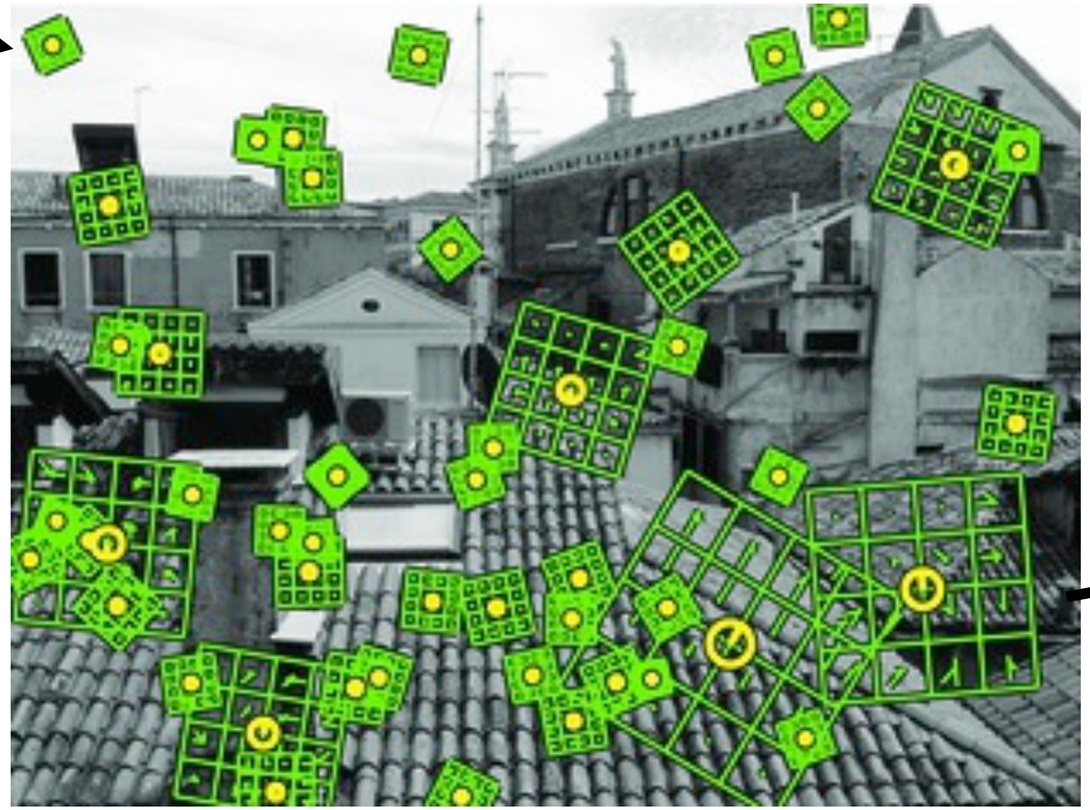
SIFT



Learned Descriptors

Invariant Local Features

Feature Detection



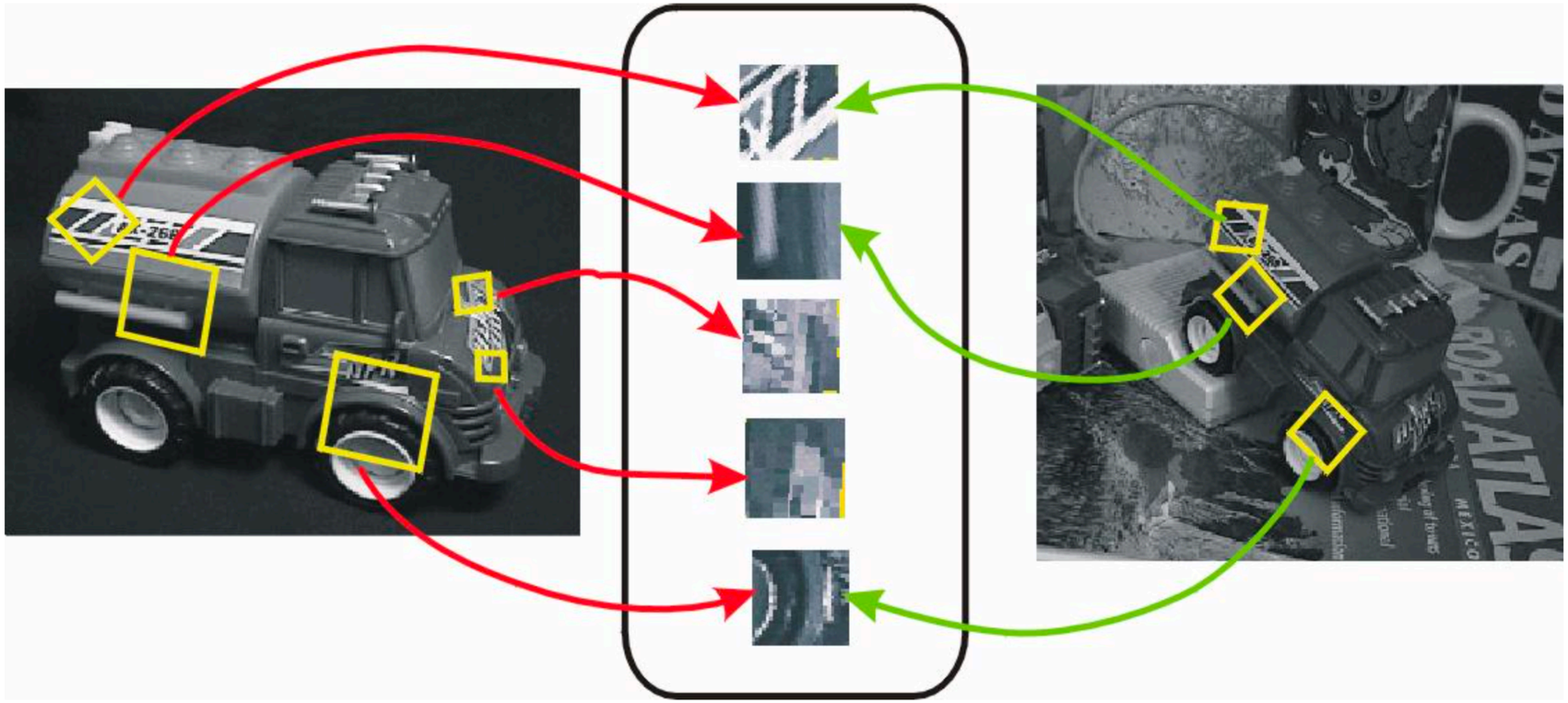
$$\begin{bmatrix} 12 \\ 105 \\ 50 \\ 198 \\ 125 \\ 15 \\ 142 \\ \dots \end{bmatrix}$$

Feature Descriptor

- **Goal**: for each detected point extract a vector of numbers (descriptor) that is as much as possible invariant to the imaging conditions — geometric distortions, photometric changes, etc.

Invariant Local Features

A good starting point is to transform into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Local Coordinate frame
a.k.a. Canonical Frame

Advantages of Invariant Local Features

Locality: features are local, so robust to occlusion and clutter (no prior segmentation)

Distinctiveness: individual features can be matched to a large database of objects

Quantity: many features can be generated for even small objects

Efficiency: close to real-time performance

Object **Recognition** with Invariant Features

Task: Identify objects or scenes and determine their pose and model parameters

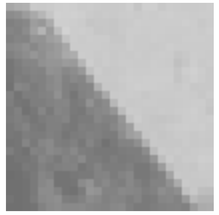
Applications:

— Industrial automation and inspection
— Mobile robots, toys, user interfaces
— Location recognition
— Digital camera panoramas
— 3D scene modeling, augmented reality

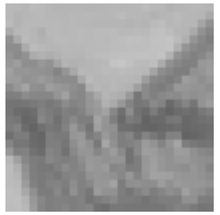
Image Structure

- What kind of structures are present in the image locally?

0D Structure: not useful for matching



1D Structure: edge, can be localised in one direction, subject to the "aperture problem"

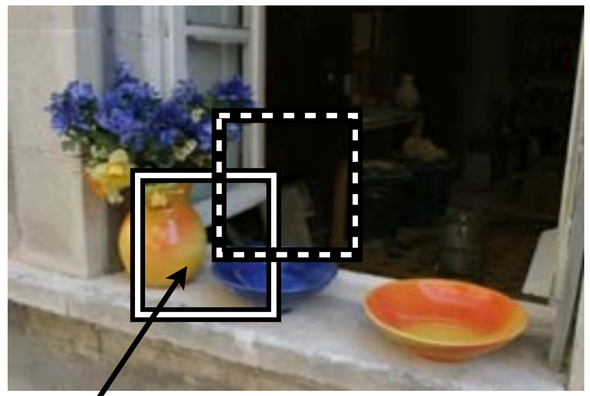


2D Structure: corner, or interest point, can be localised in both directions, good for matching

Edge detectors find contours (1D structure), **Corner** or **Interest point** detectors find points with 2D structure.

Local SSD Function

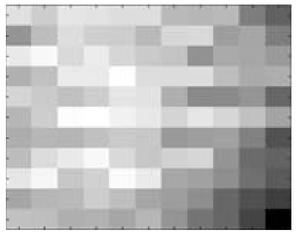
- Consider the sum squared difference (SSD) of a patch with its local neighbourhood

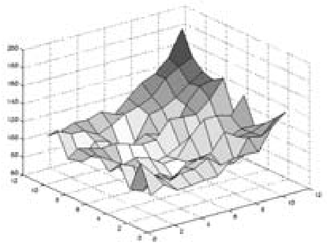


$$\text{SSD} = \sum_{\mathcal{R}} |I(\mathbf{x}) - I(\mathbf{x} + \Delta\mathbf{x})|^2$$

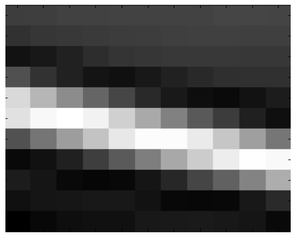
Local SSD Function

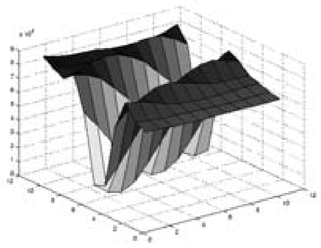
- Consider the local SSD function for different patches



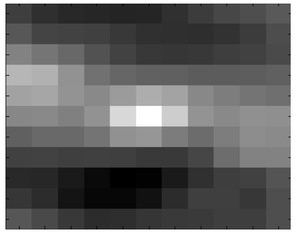


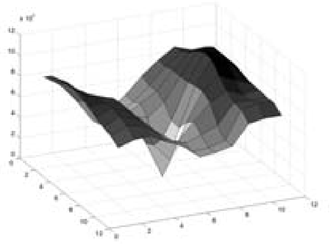
High similarity locally





High similarity along the edge

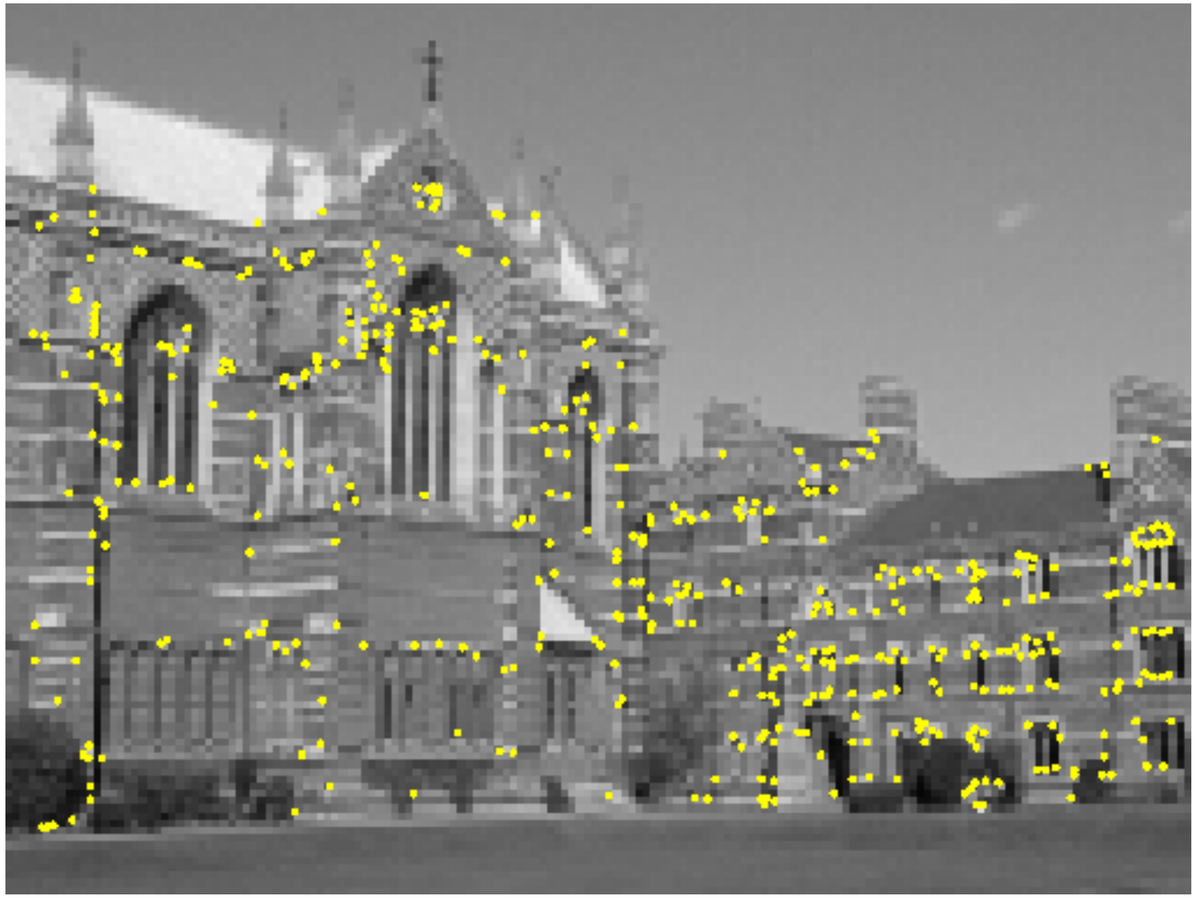


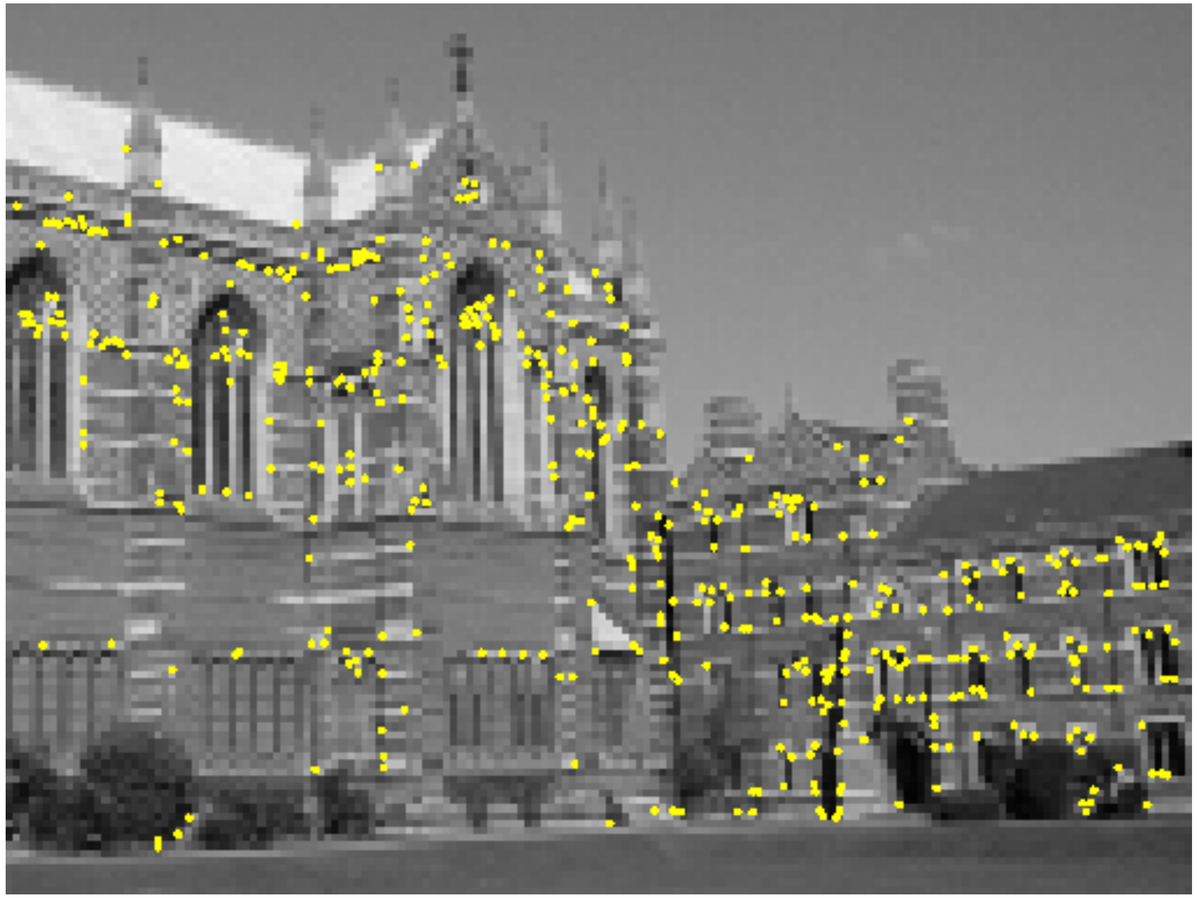


Clear peak in similarity function

Harris Corners

- Harris corners are peaks of a local similarity function





Lecture 9: Re-cap (compute the covariance matrix)

Sum over small region around the corner

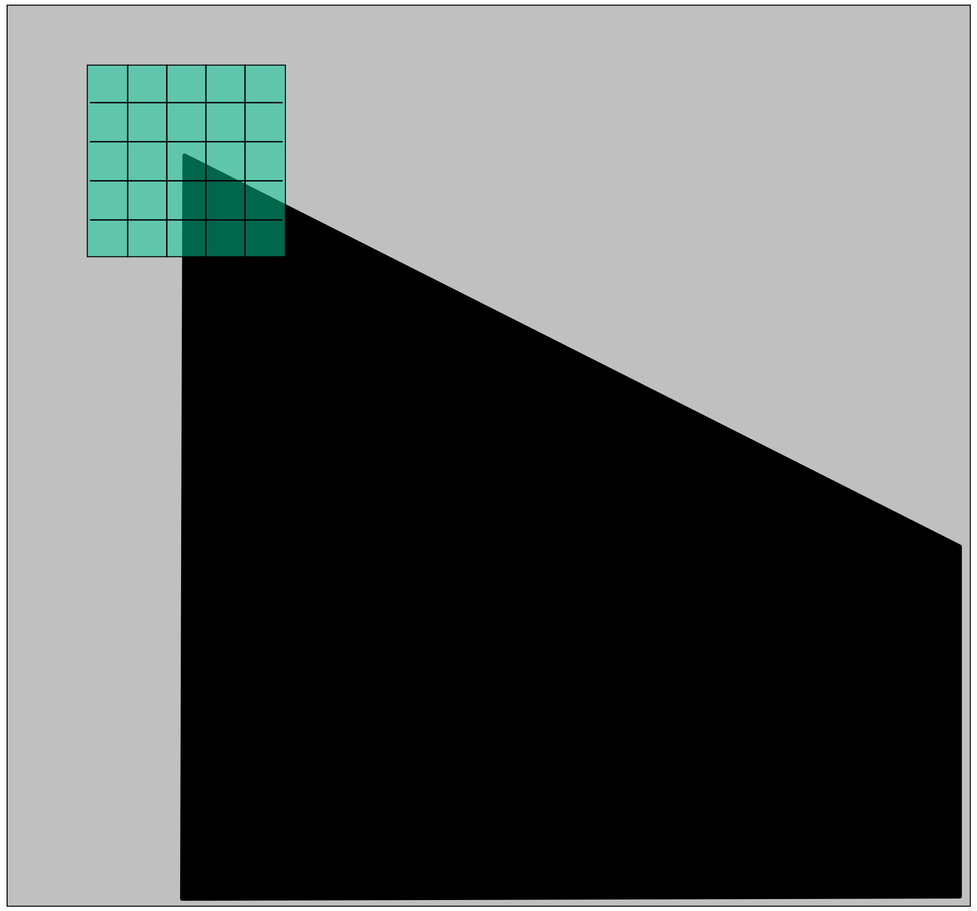
Gradient with respect to x, times gradient with respect to y

$$C = \begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$

Matrix is **symmetric**

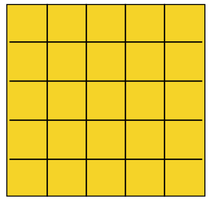
1. Compute **image gradients** over a small region

(not just a single pixel)



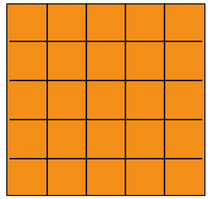
array of x gradients

$$I_x = \frac{\partial I}{\partial x}$$



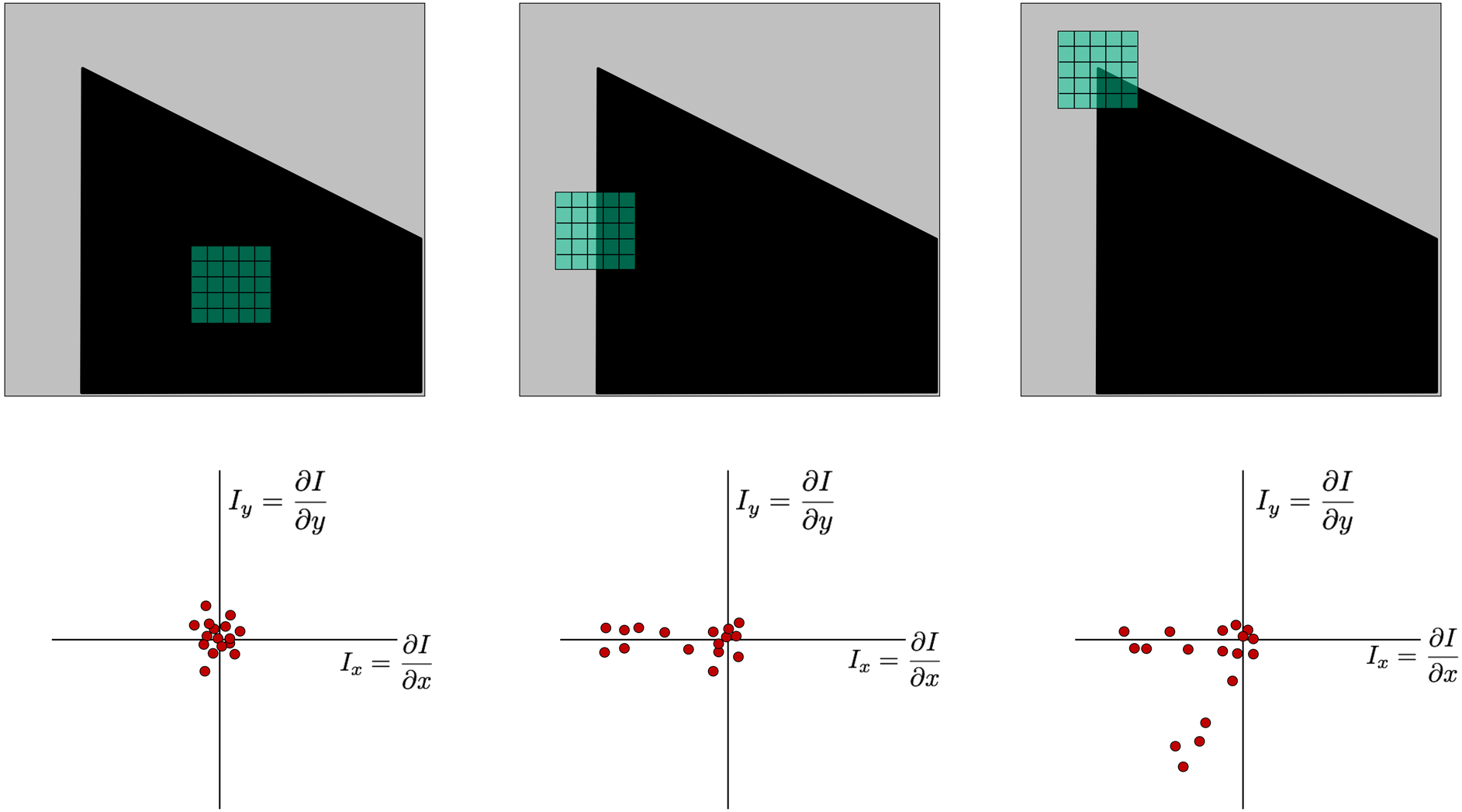
array of y gradients

$$I_y = \frac{\partial I}{\partial y}$$



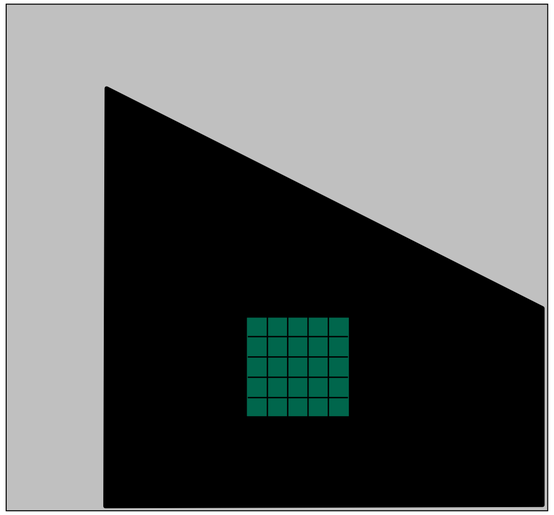
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

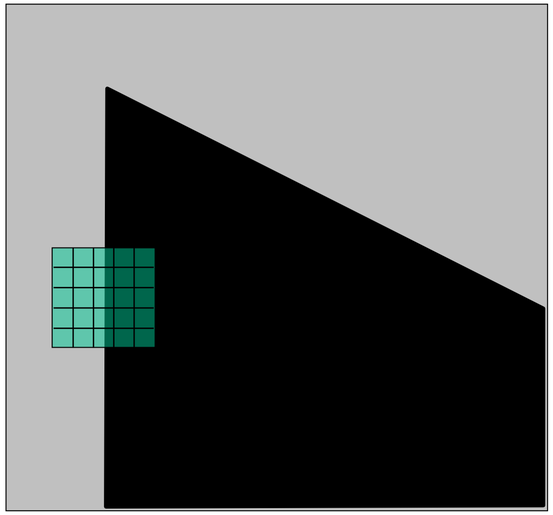
What Does a **Distribution** Tells You About the **Region**?

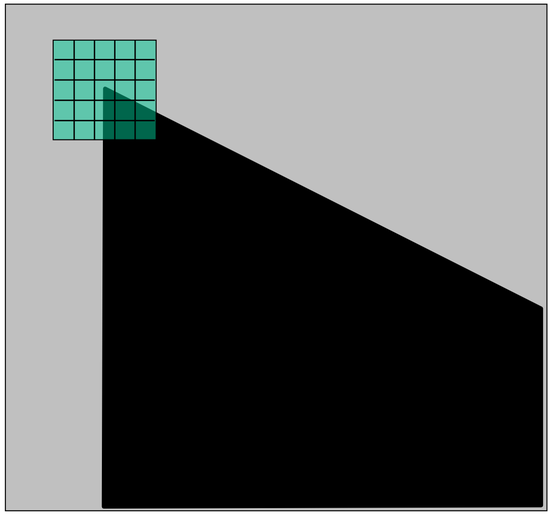


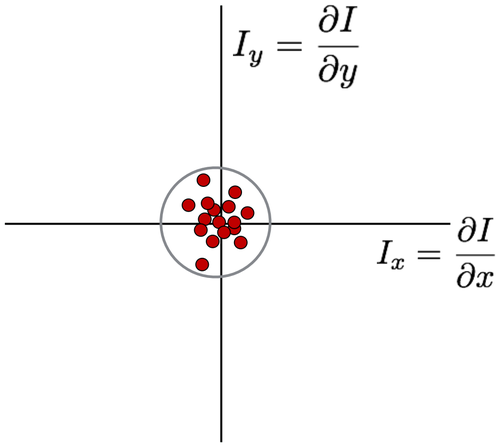
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

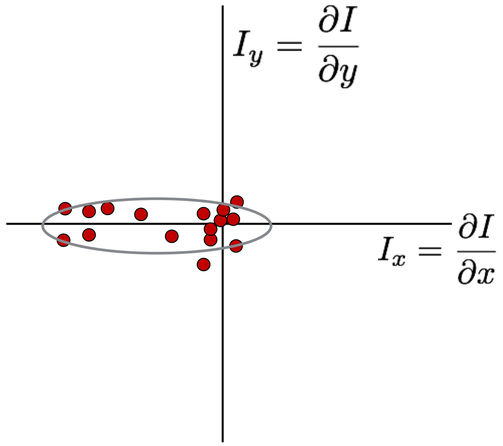
What Does a **Distribution** Tells You About the **Region**?

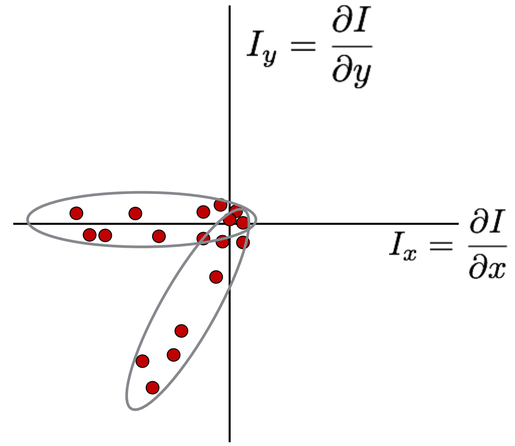












Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Harris Corner Detection Review

- Filter image with **Gaussian**
- Compute magnitude of the x and y **gradients** at each pixel
- Construct C in a window around each pixel
  - Harris uses a **Gaussian window**
- Compute the Harris corner strength function $\det(C) - \kappa \text{trace}^2(C)$
- This requires that both λ's (eigenvalues of C) are large

MSERS

- Maximally Stable Extremal Regions

- Find regions of high contrast using a watershed approach

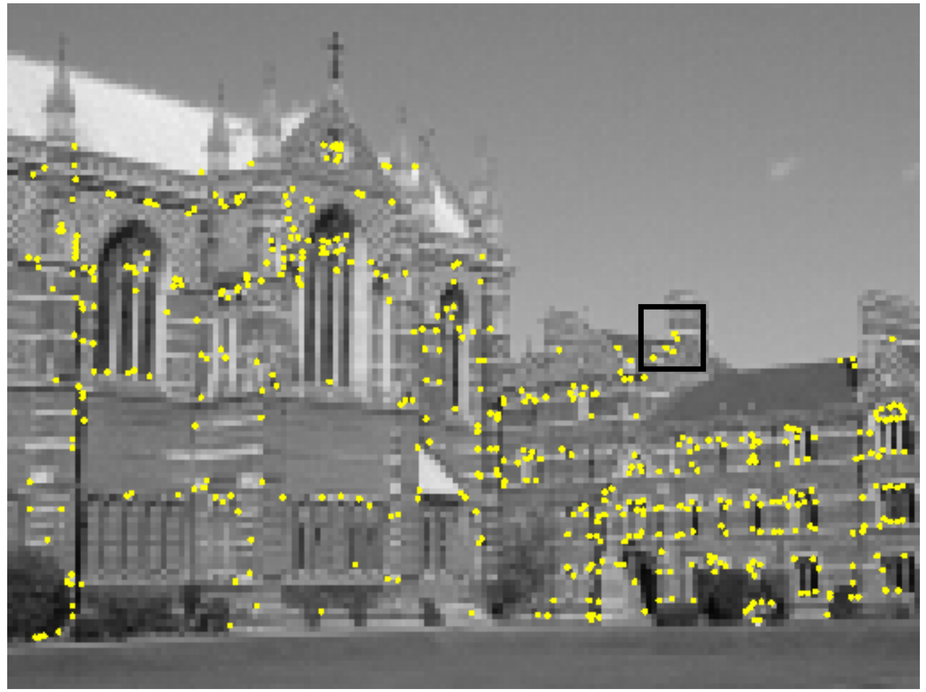

13.1

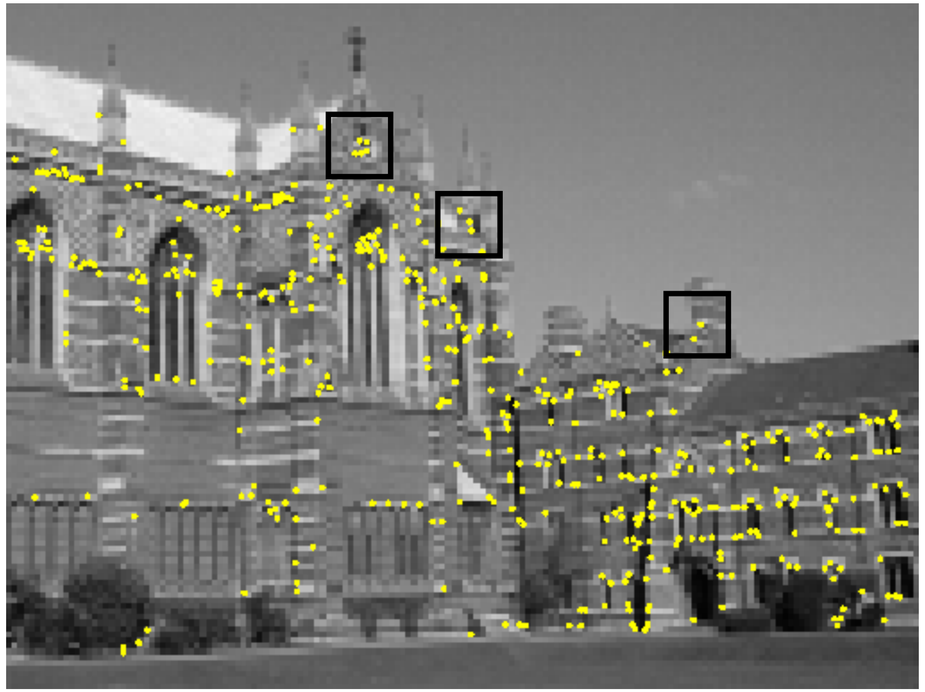
MSERS are stable (small change) over a large range of thresholds

[Matas et al 2002]

Corner Matching

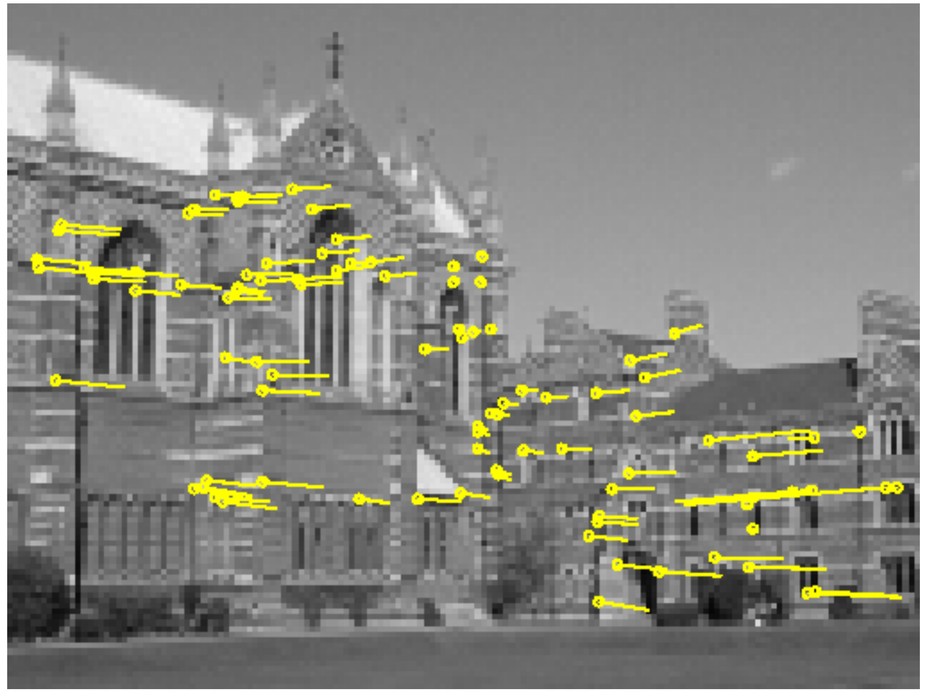
- A simple approach to correspondence is to match corners between images using normalised correlation or SSD





Harris Corners

- Corners matched using correlation



99 inliers

89 outliers

[Zhang, Deriche, Faugeras, Luong 1995, Beardsley, Torr, Zisserman 1996]

Breaking Correlation

- Correlation/SSD works well when the images are quite similar (e.g., tracking in frames of a video)
- However, it is easily broken by simple image transforms, e.g.,

Original Rotation Scale

- These transformations are very common in imaging, so we would like feature matching to be **invariant** to them

Motivation: Template Matching

When might **template matching fail**?

— Different scales 

— Different orientation 

— Lighting conditions 

— Left vs. Right hand 

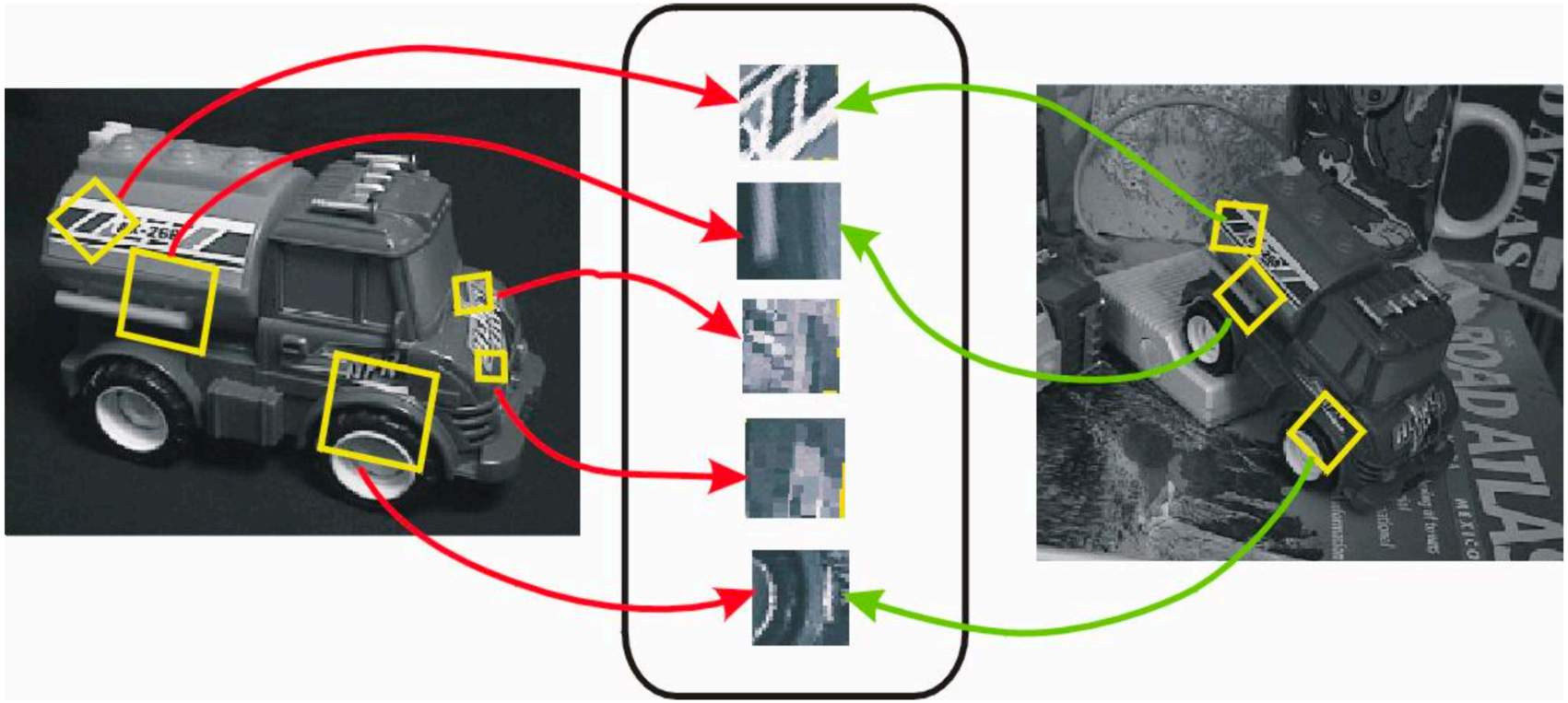
— Partial Occlusions 

— Different Perspective

— Motion / blur

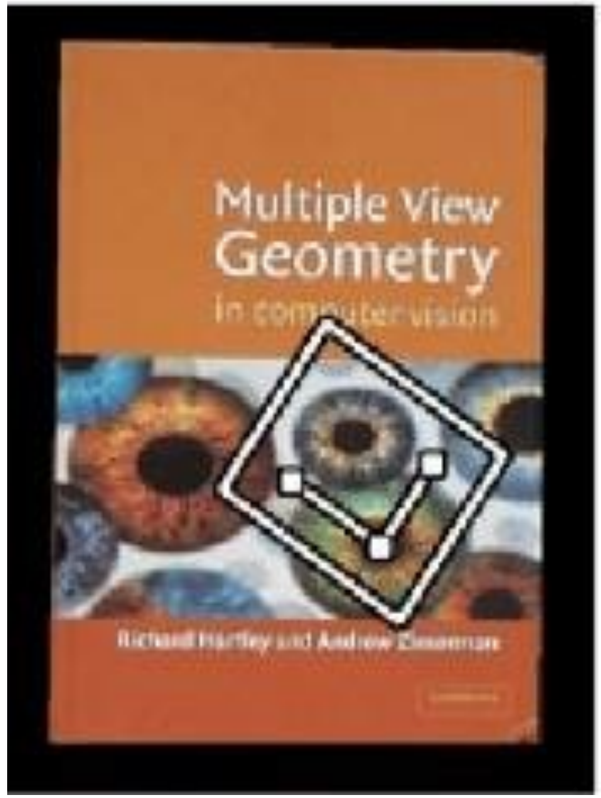
Invariant Local Features

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



Local Coordinate frame
a.k.a. Canonical Frame

Local **Coordinate Frames**



One approach is to use a set of nearby points to establish a local coordinate frame, and find a corresponding set of points in a new image

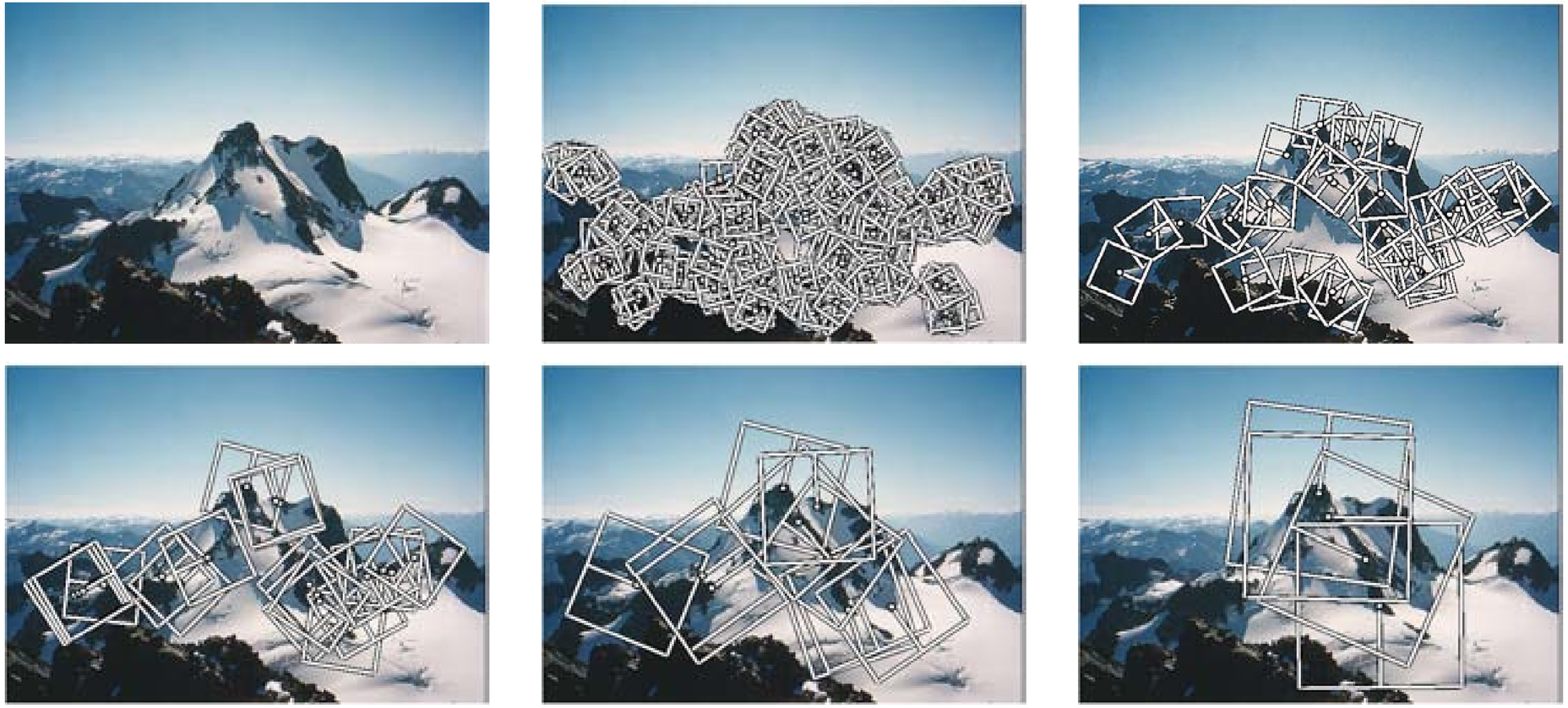
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Local Coordinate Frame

- If the local coordinate frame follows the surface transformation (covariant), the sampling of the image in that frame is invariant

Detecting Scale/Orientation

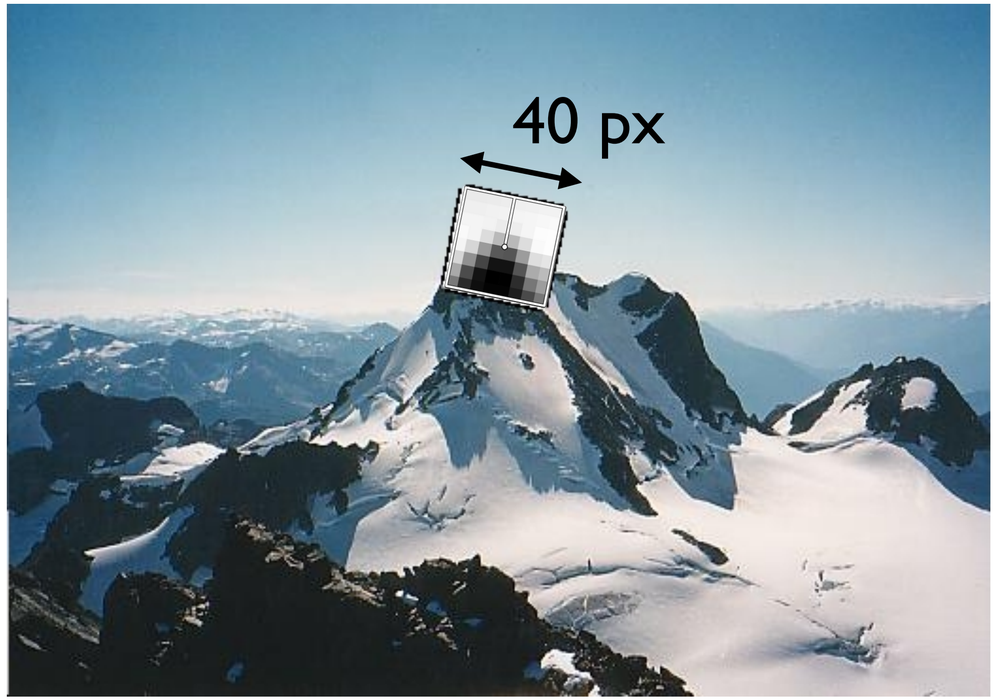
- Another method to establish a local coordinate frame is to detect a local scale and orientation for each feature point

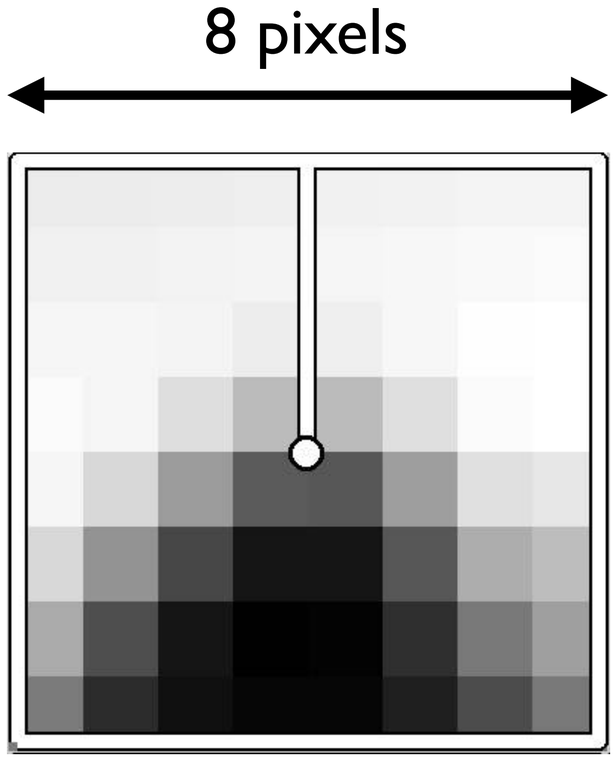


e.g., extract Harris at multiple scales and align to the local gradient

Detecting Scale/Orientation

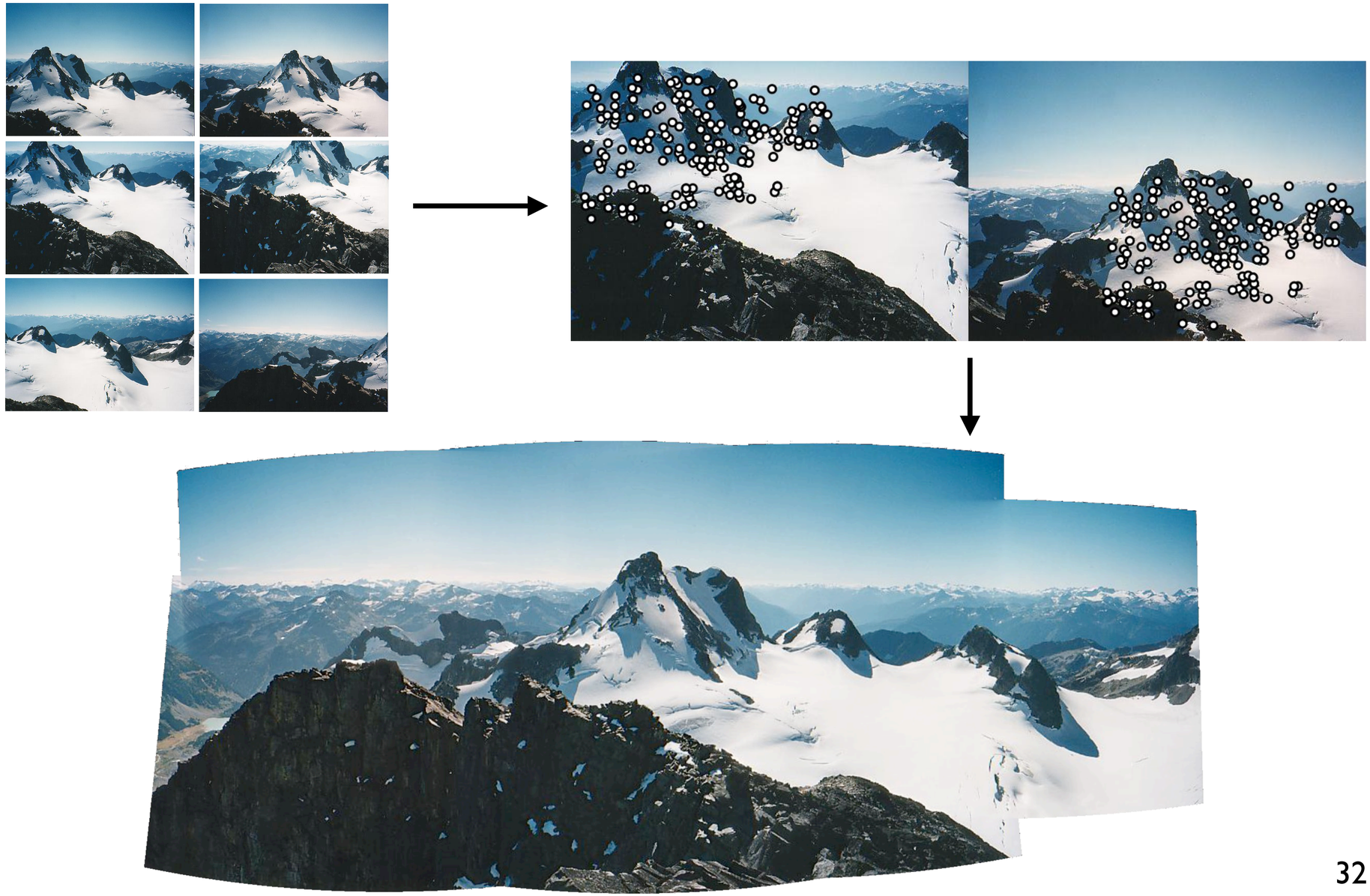
- Patch matching can be improved by using scale/orientation aligned frame and using brightness normalisation





Sampling at a coarser scale than detection further improves robustness

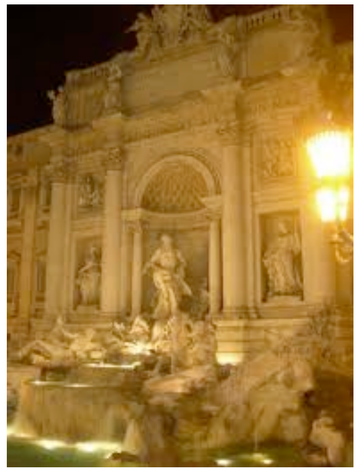
Panorama Alignment



Wide Baseline Matching

- Patch-based matching works well for short baselines, but fails for large changes in scale, rotation or 3D viewpoint




What factors cause differences between these images?

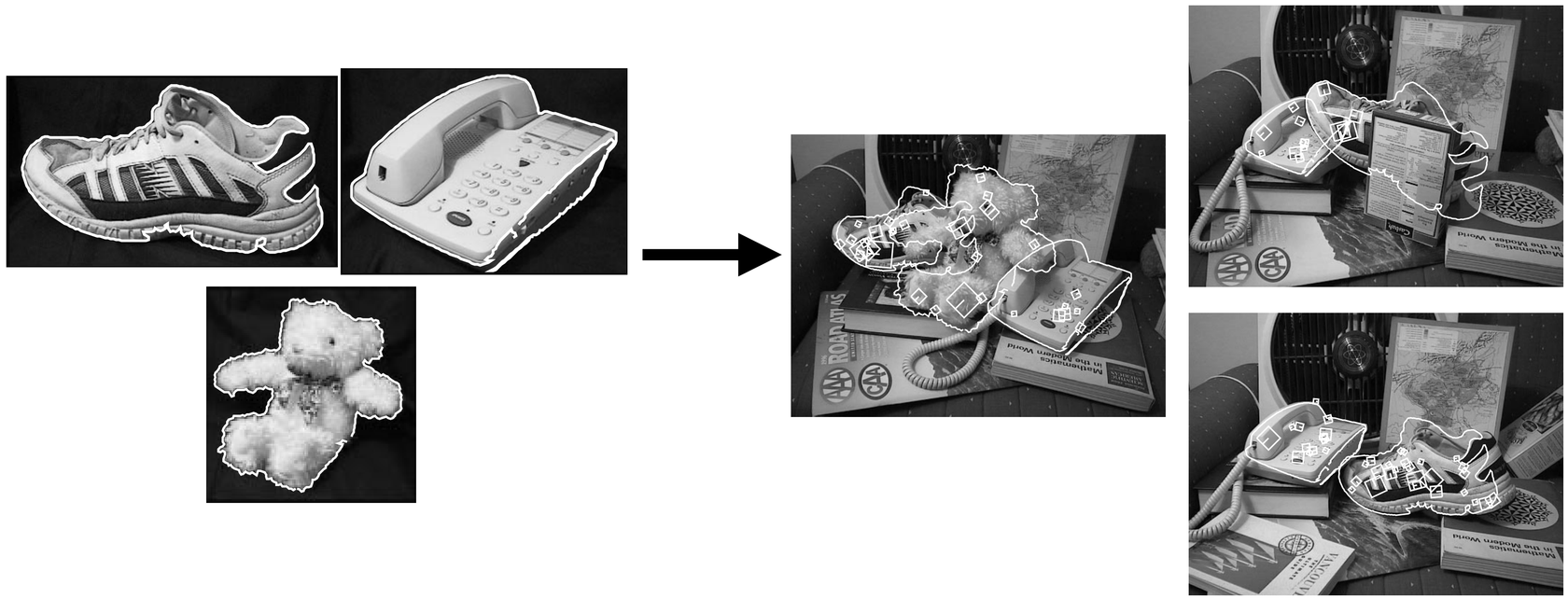
Wide Baseline Matching

- We would like to match patches despite these changes

What features of the local patch are **invariant**?

Scale Invariant Feature Transform

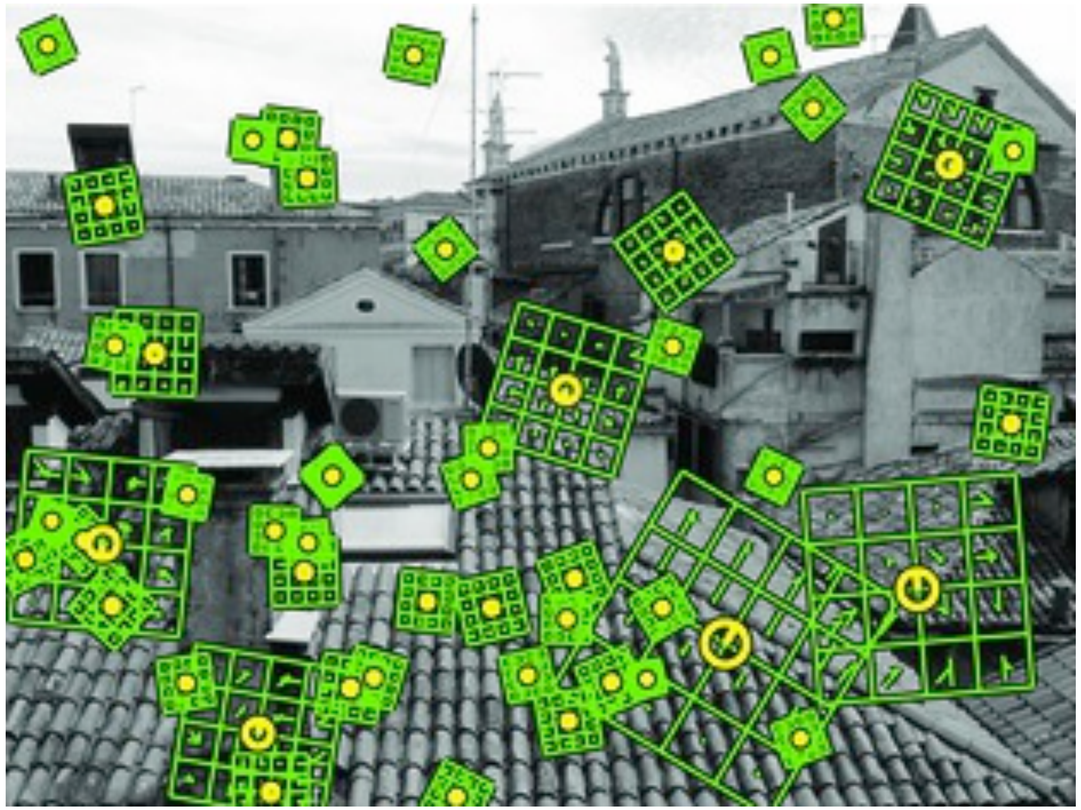
- A detector and descriptor designed for object recognition



- SIFT features are invariant to translation, rotation and scale and slowly varying under perspective and 3D distortion
- Variants widely used in object recognition, image search etc.

[Lowe 1999]

Scale Invariant Feature Transform

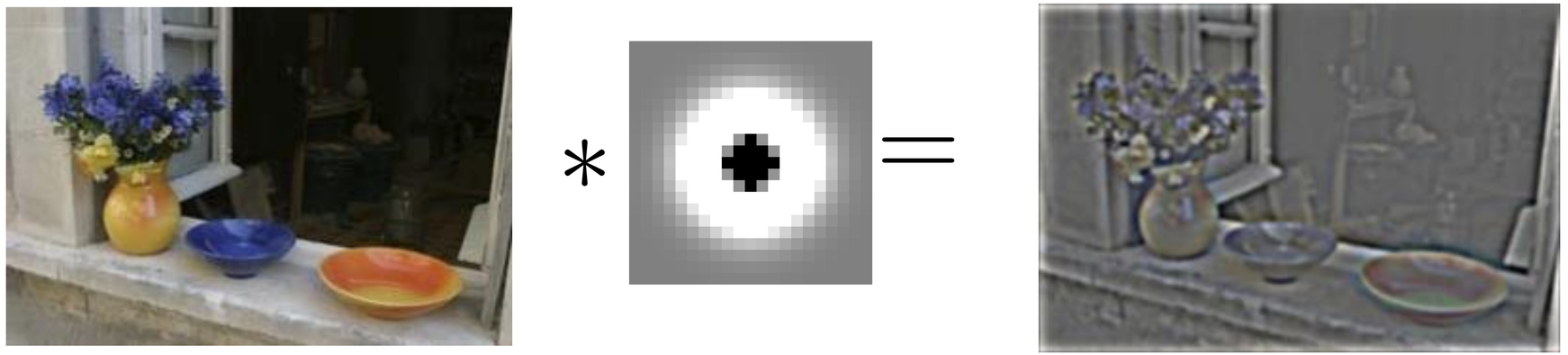


[vlfeat.org]

- Scale invariant detection and local orientation estimation
- **Edge based** representation that is robust to local shifting of edges (parallax and/or stretch)

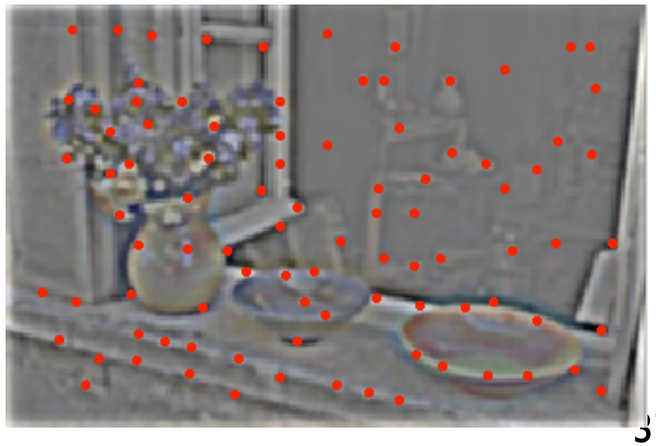
SIFT Detection

- Convolve with a centre-surround Laplacian / DoG filter



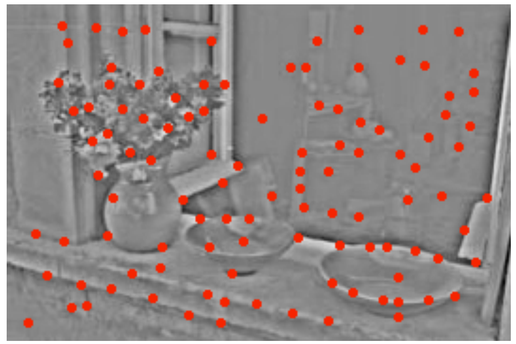
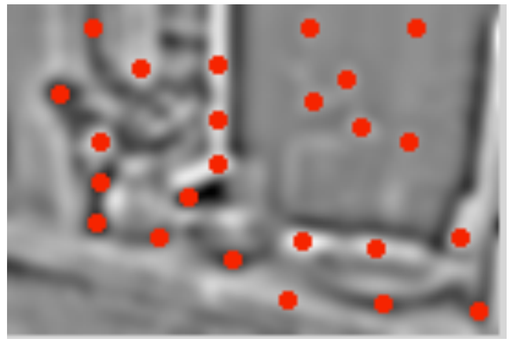
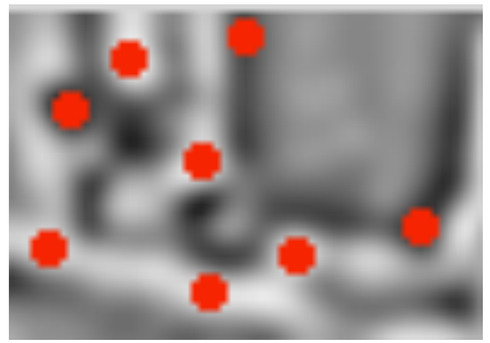
- Find local-maxima of the centre surround response

Non-maximal suppression:
These points are maxima in
a 10 pixel radius



SIFT Detection

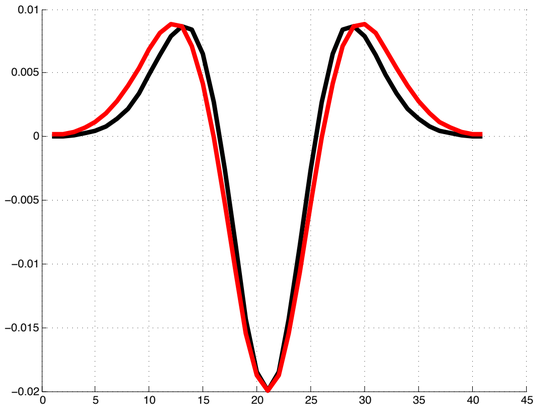
- DoG detects blobs at scale that depends on the Gaussian standard deviation(s)

Note: DOG ≈ Laplacian of Gaussian

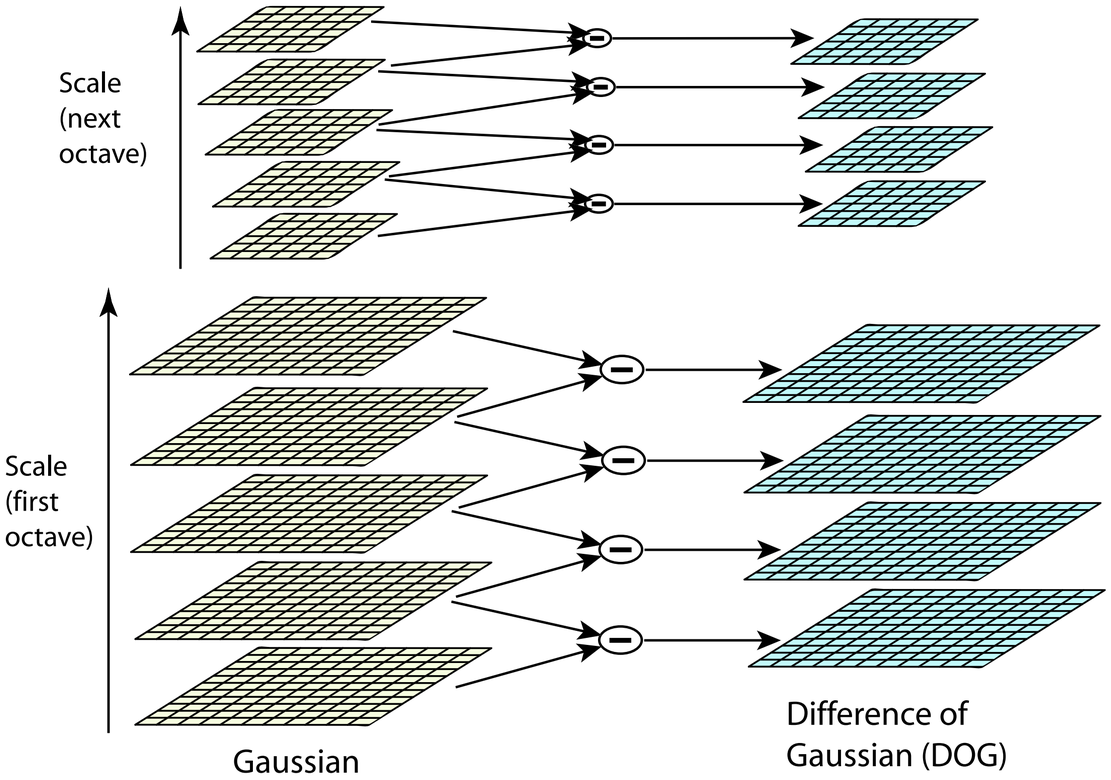
$$\text{red} = [1 \ -2 \ 1] * g(x;\ 5.0)$$

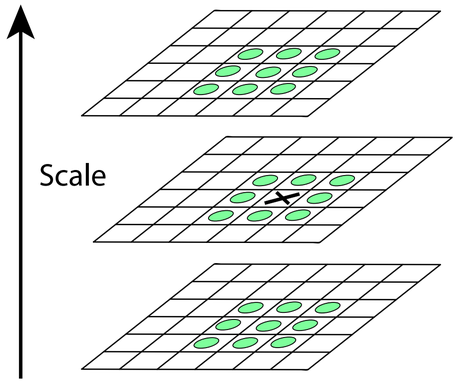
$$\text{black} = g(x;\ 5.0) - g(x;\ 4.0)$$



Scale Selection

- A DoG (Laplacian) Pyramid is formed with multiple scales per octave

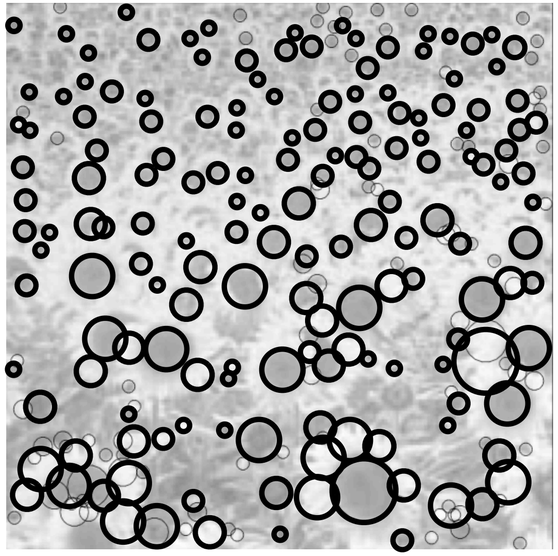


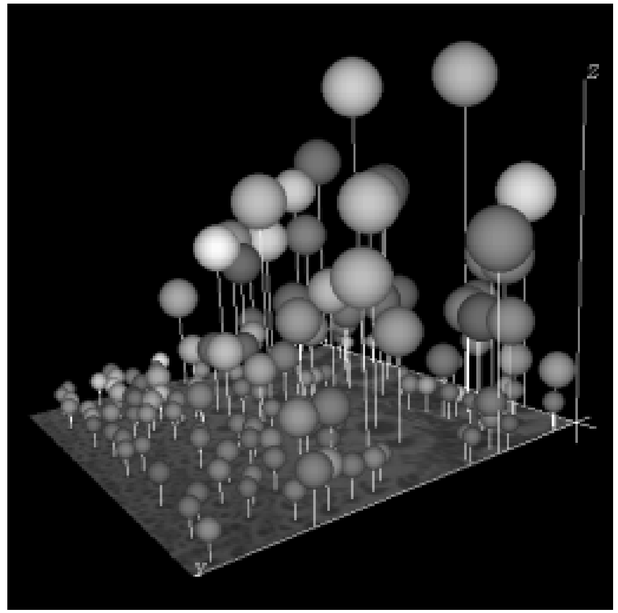


Detections are local maxima in a 3x3x3 scale-space window

Scale Selection

- Maximising the DoG function in scale as well as space performs scale selection

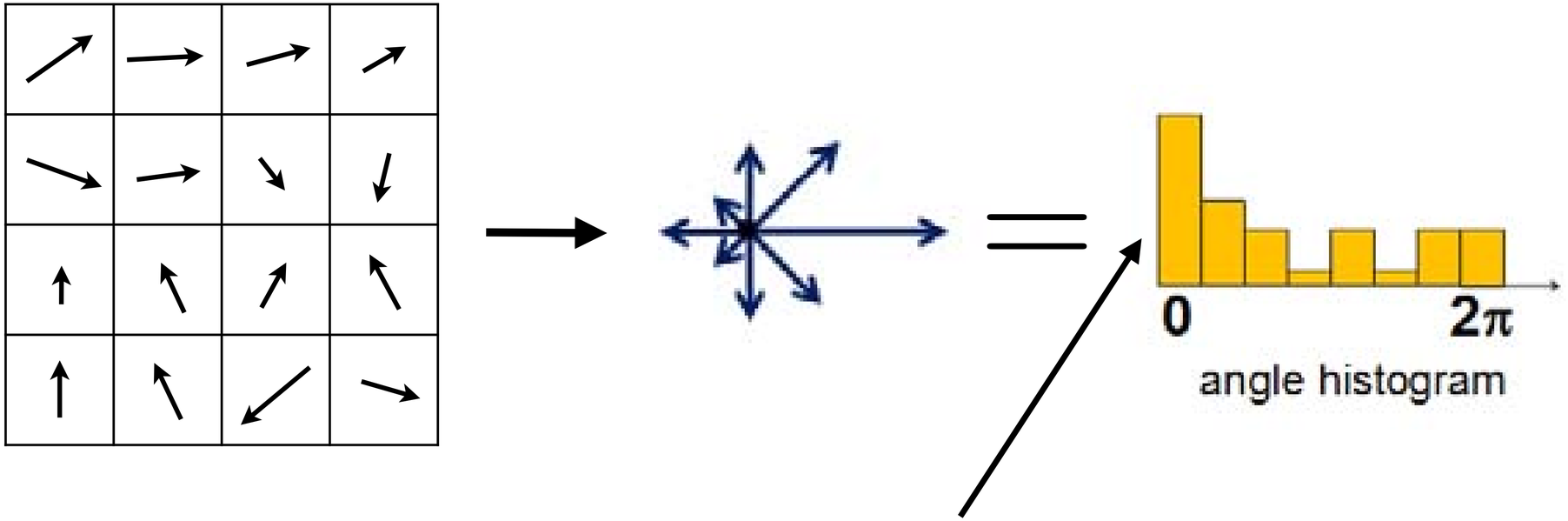




[T. Lindeberg]

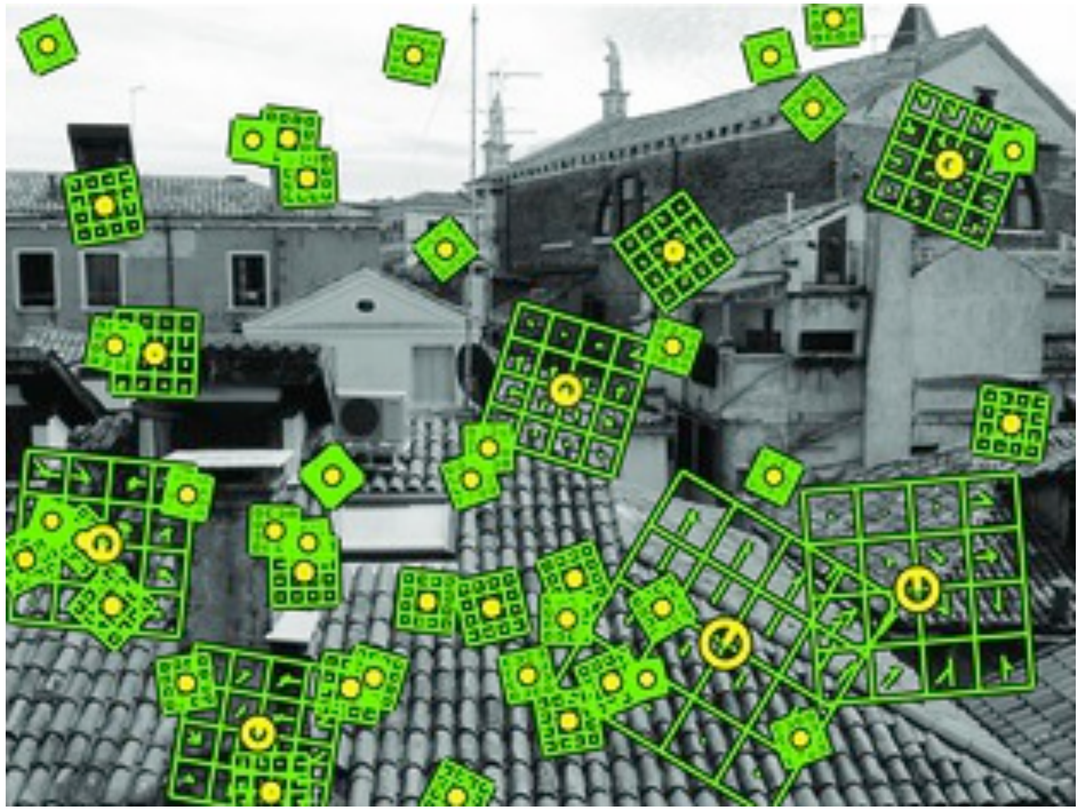
Orientation Selection

- To select a local orientation, build a histogram over orientation



Selected orientation
is peak in this histogram

SIFT Descriptor



- We selected a scale and orientation at each detection,
- Now need **descriptor** to represent the local region in a way robust to parallax, illumination change etc.

SIFT Descriptor

$$\begin{bmatrix} 12 \\ 105 \\ 50 \\ 198 \\ 125 \\ 15 \\ 142 \\ \dots \end{bmatrix}$$

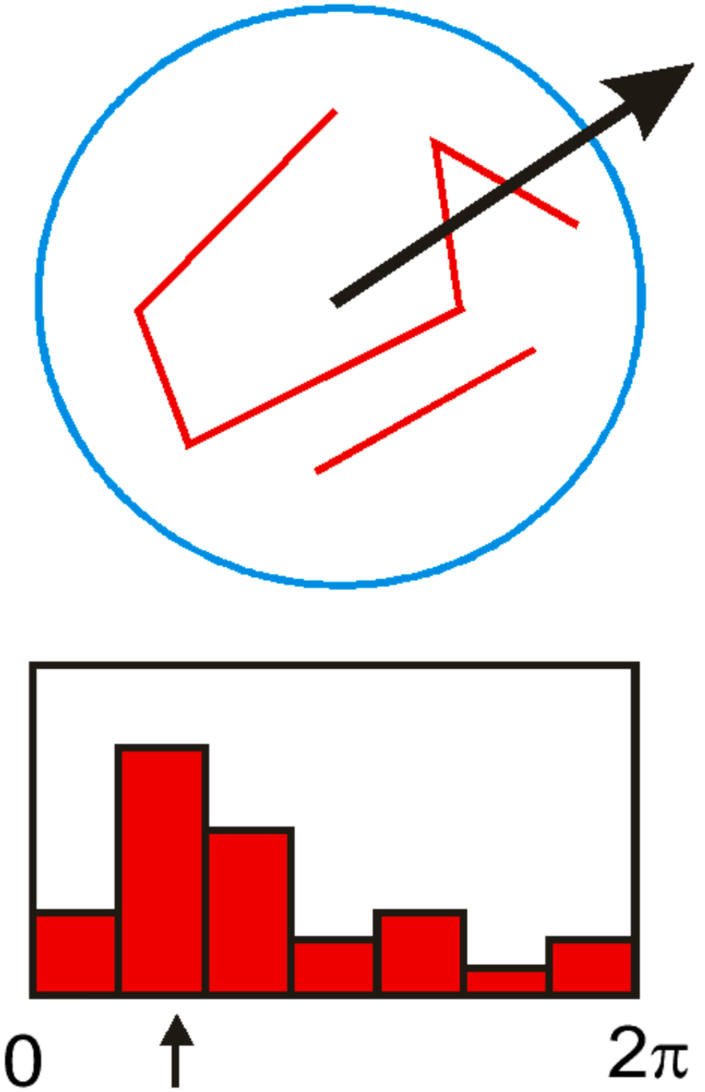
- **Goal**: for each patch extract a vector of numbers (descriptor) that is as much as possible invariant to the imaging conditions — geometric distortions, photometric changes, etc.

Gradient **Orientation Histogram**

— Create **histogram** of local gradient directions computed at selected scale

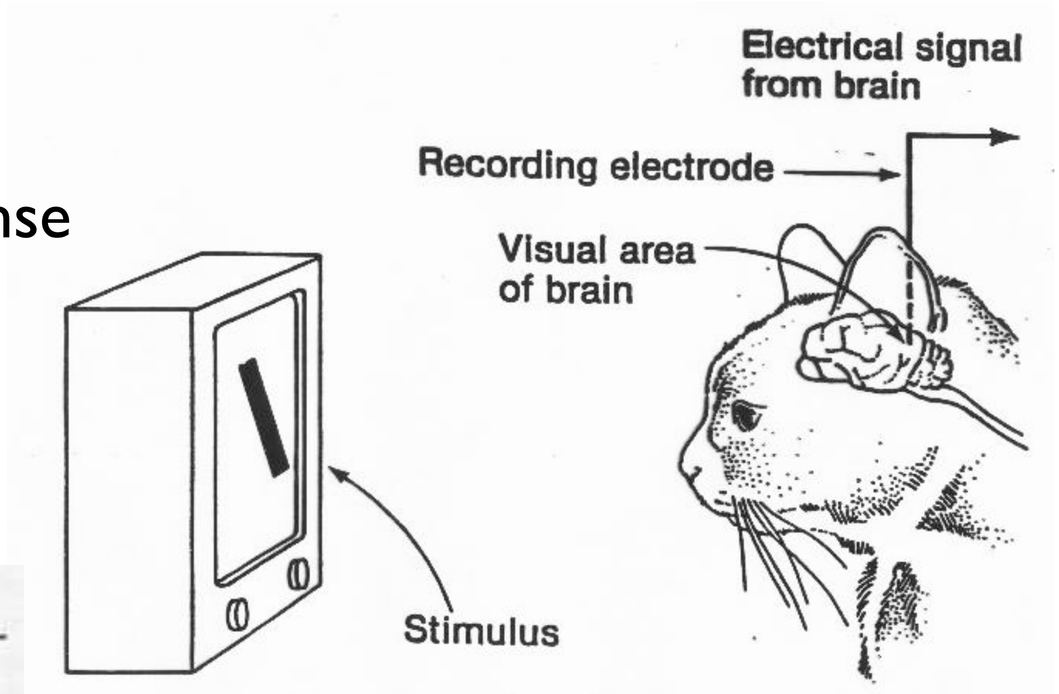
— Peak of histogram is used to assign **canonical orientation**

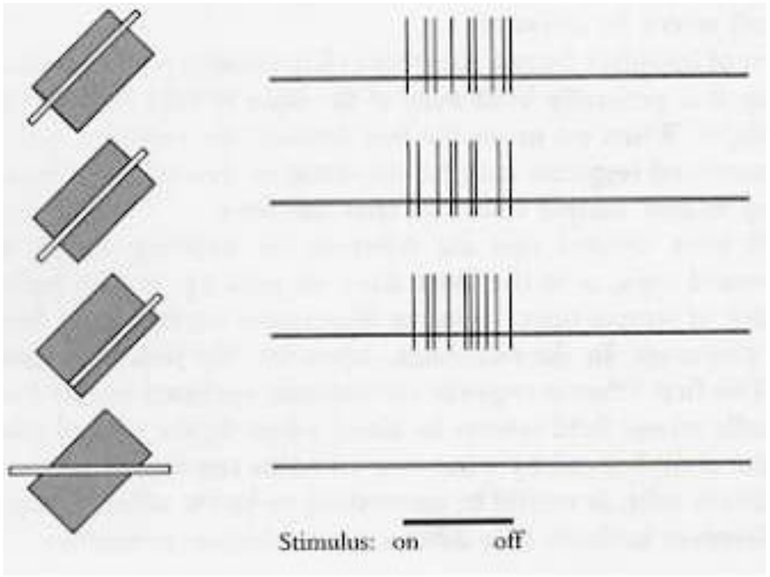
— Multiple gradient orientation histograms are used to form the **descriptor**



Simple + Complex Cells in V1

- Neuroscientists have investigated the response of cells in the primary visual cortex

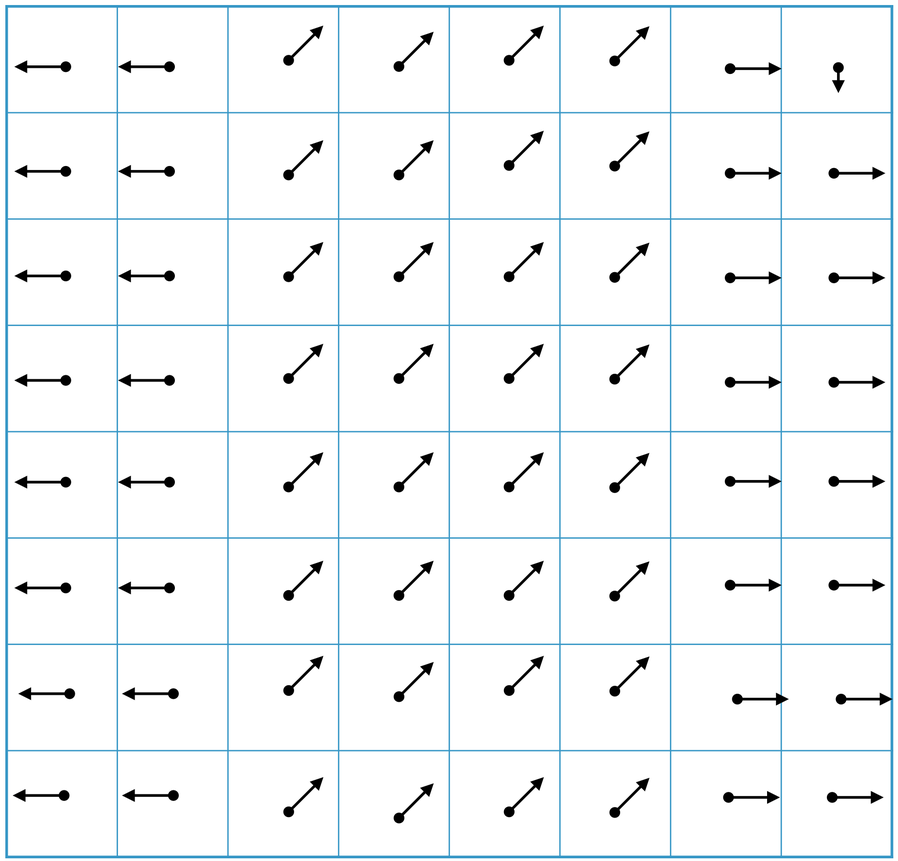




- "Complex Cells" in V1 respond over a range of positions but are highly sensitive to orientation

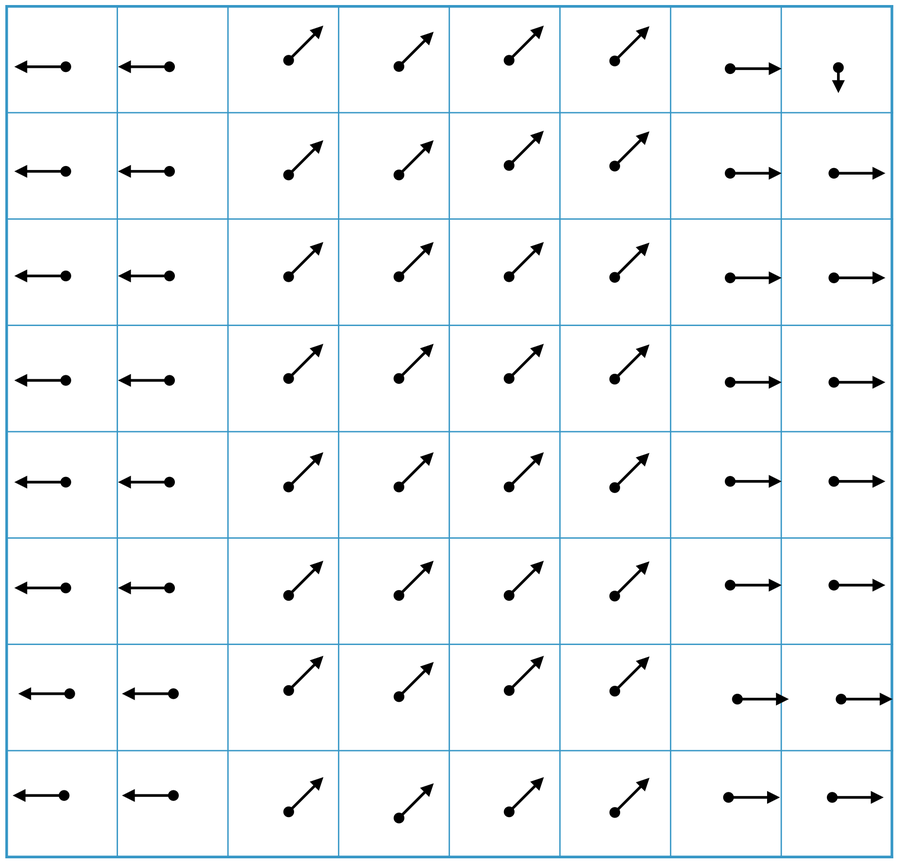
[Hubel and Wiesel]

Gradient **Orientation Histogram**

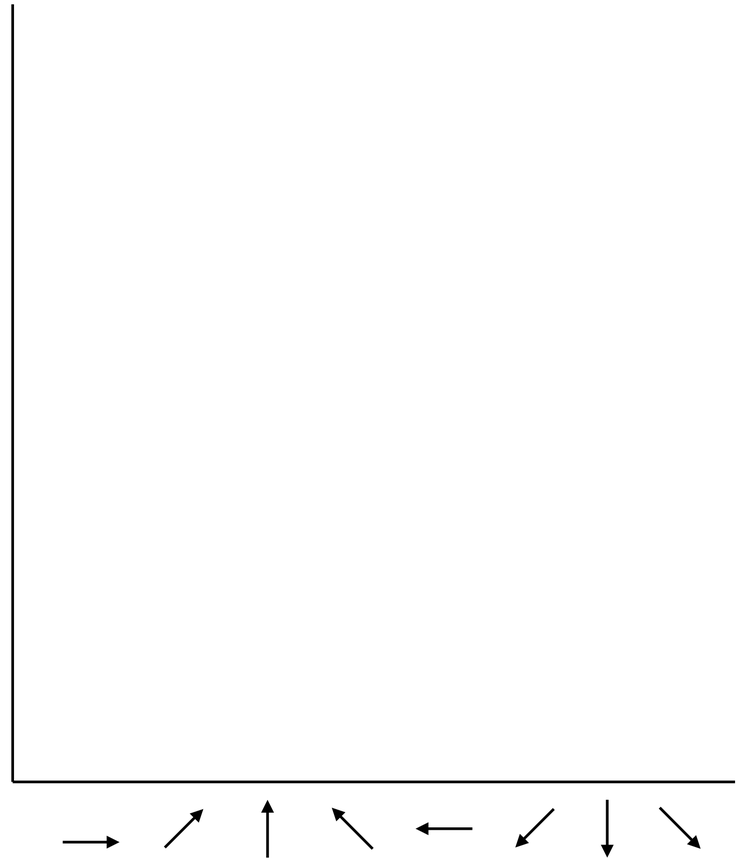


Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)

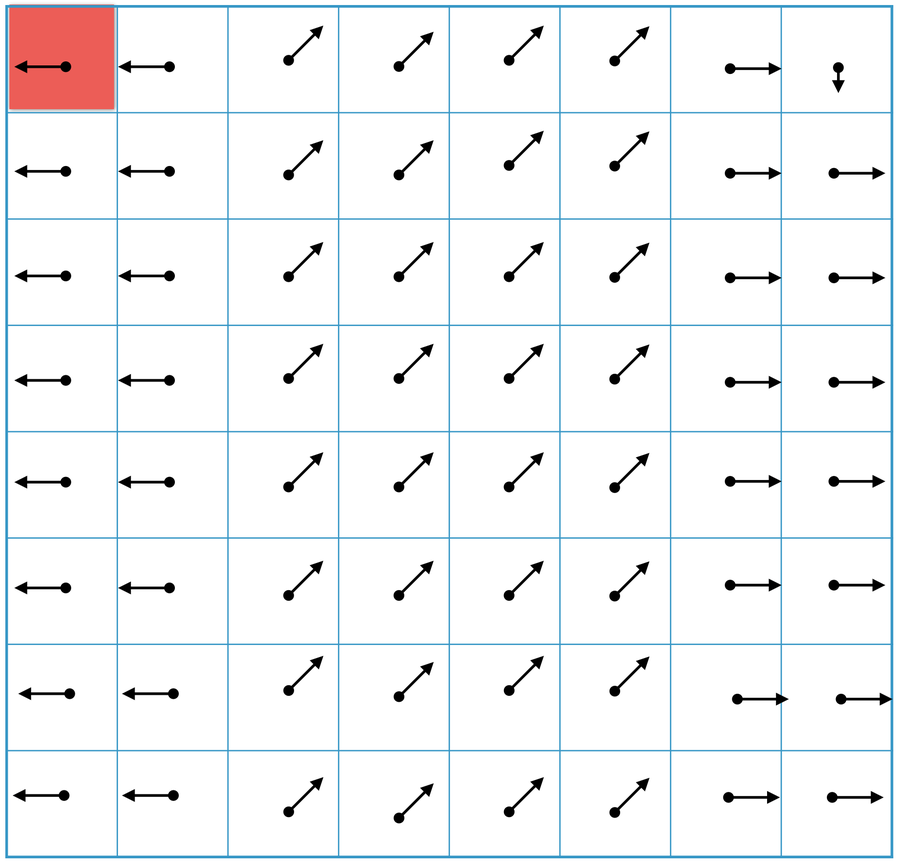
Gradient **Orientation Histogram**

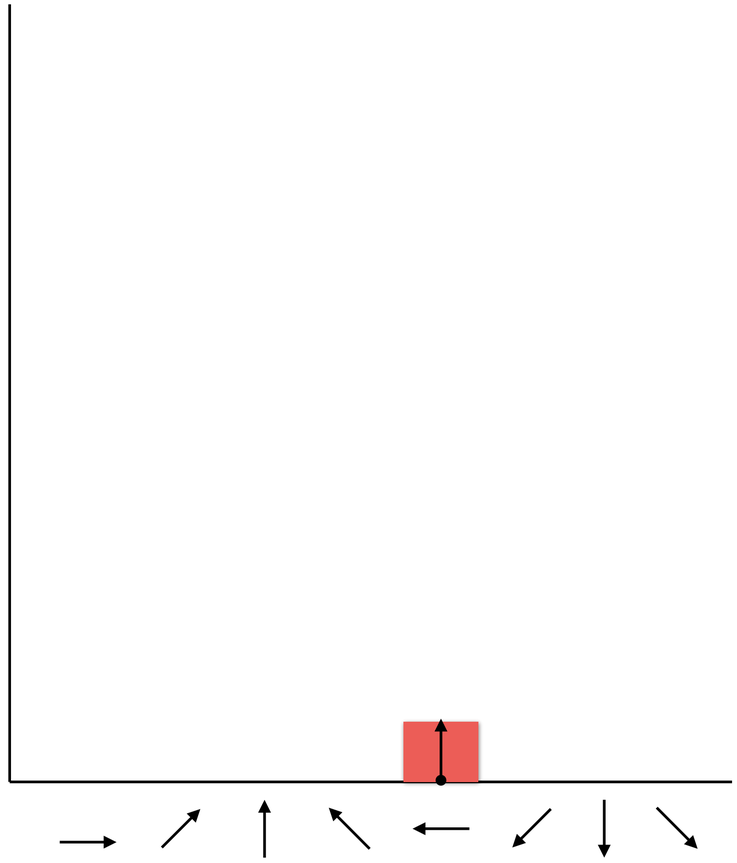


Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)



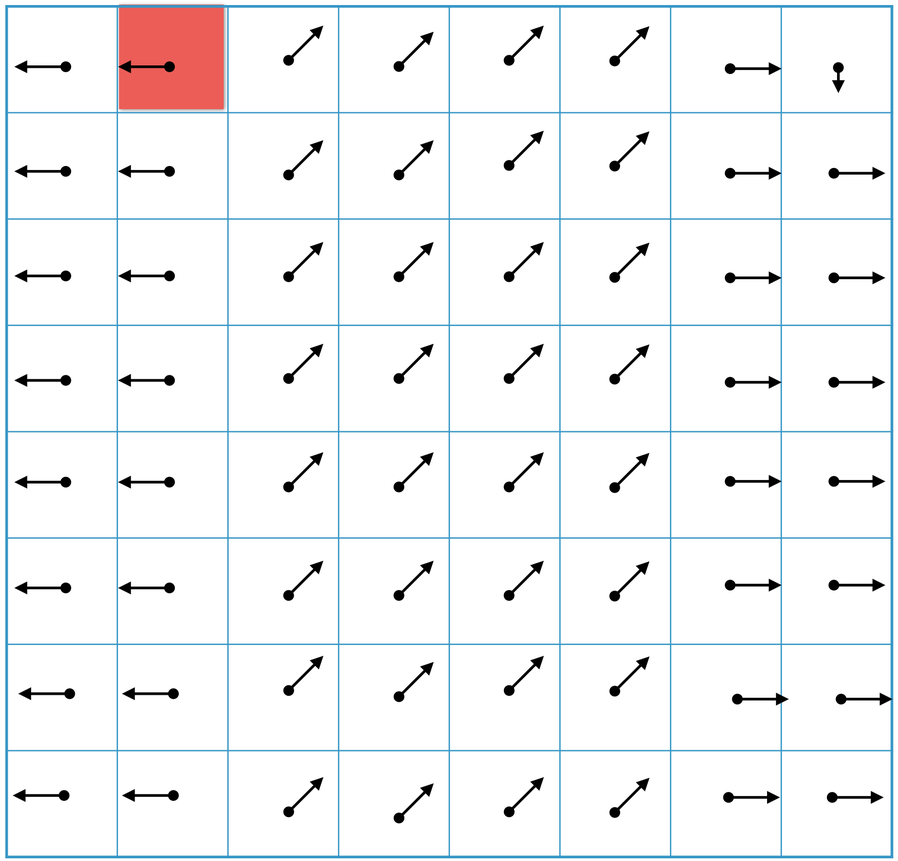
Gradient **Orientation Histogram**

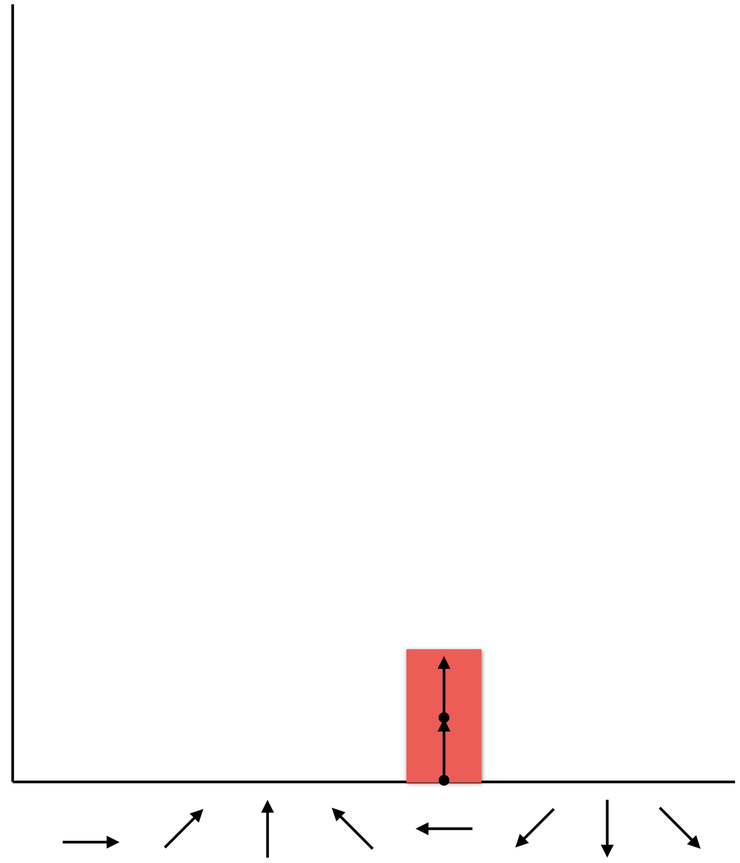




Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)

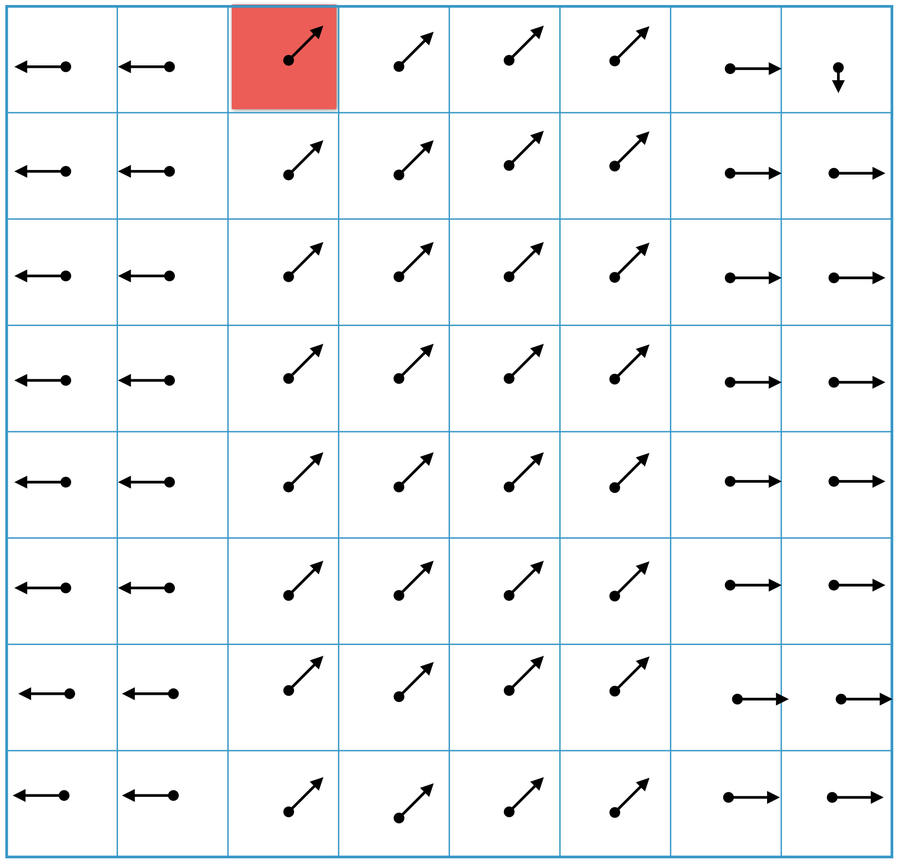
Gradient **Orientation Histogram**

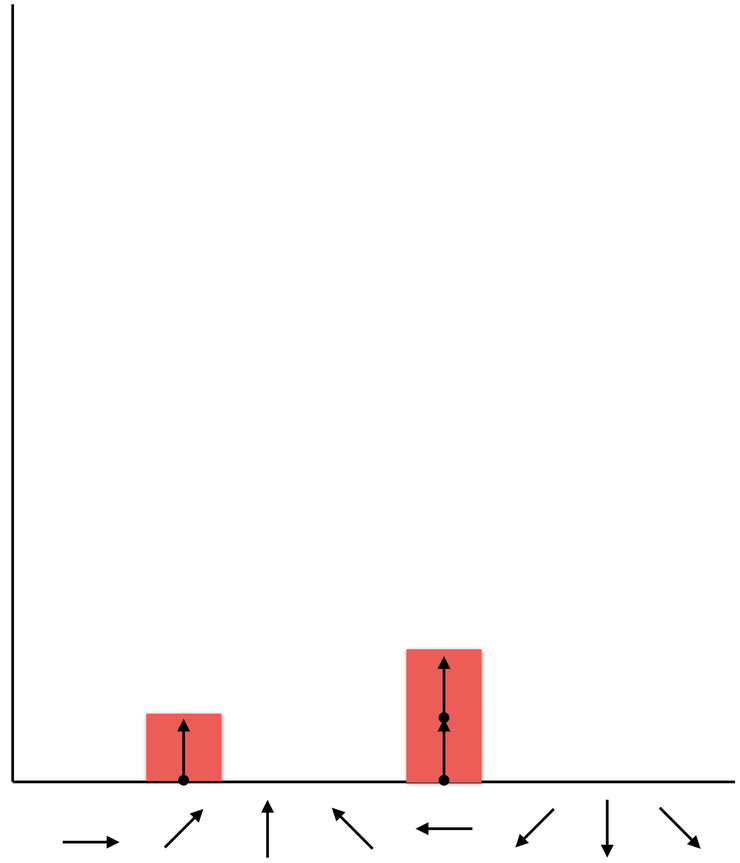




Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)

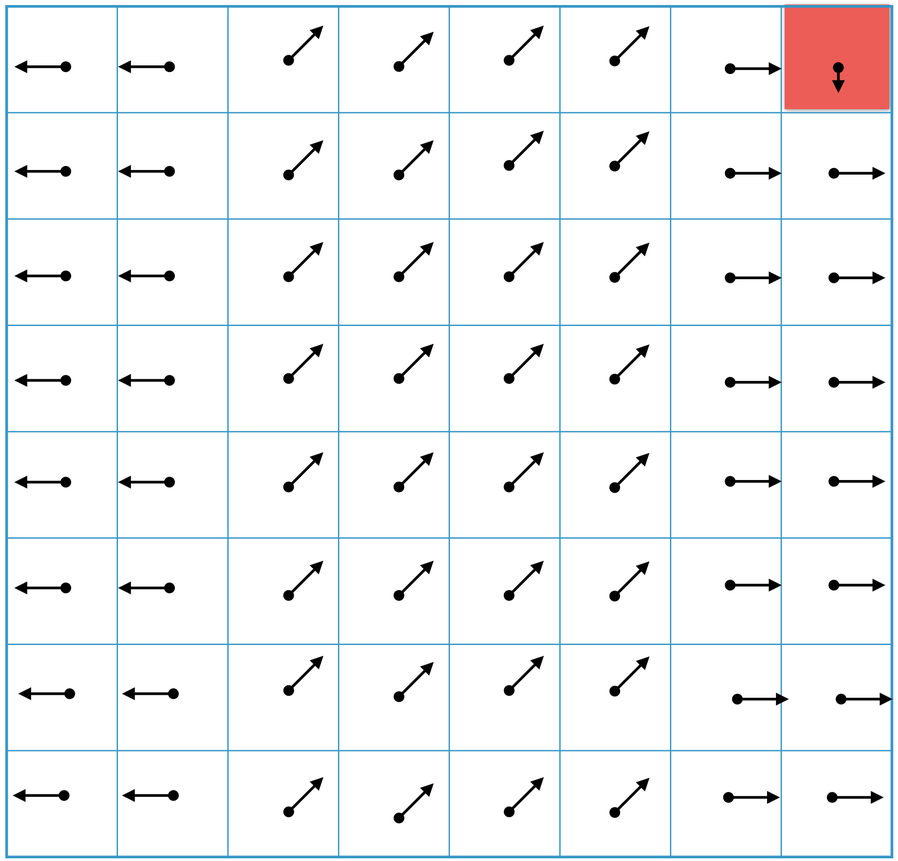
Gradient **Orientation Histogram**

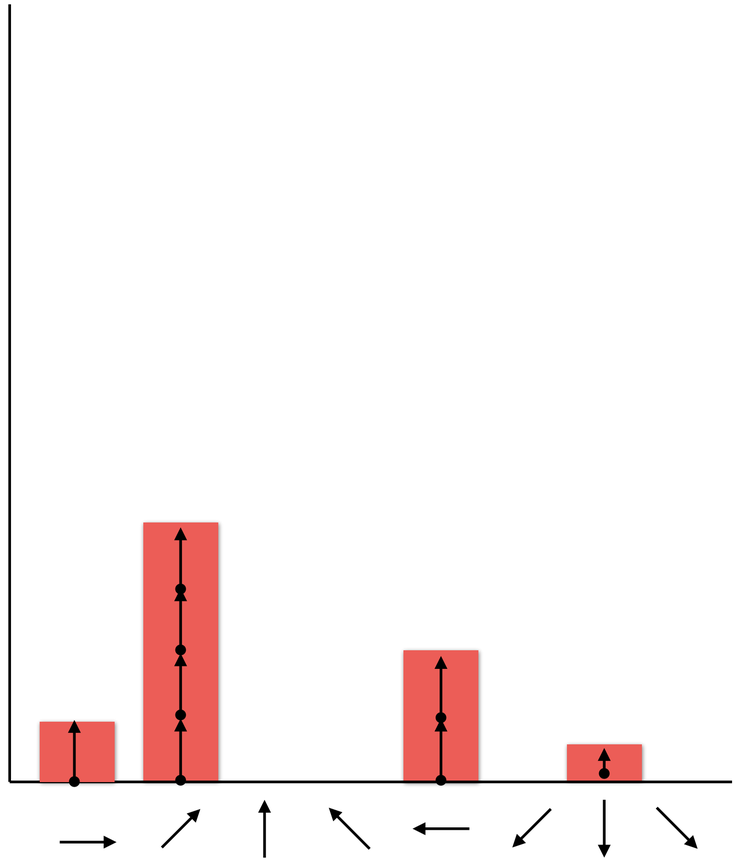




Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)

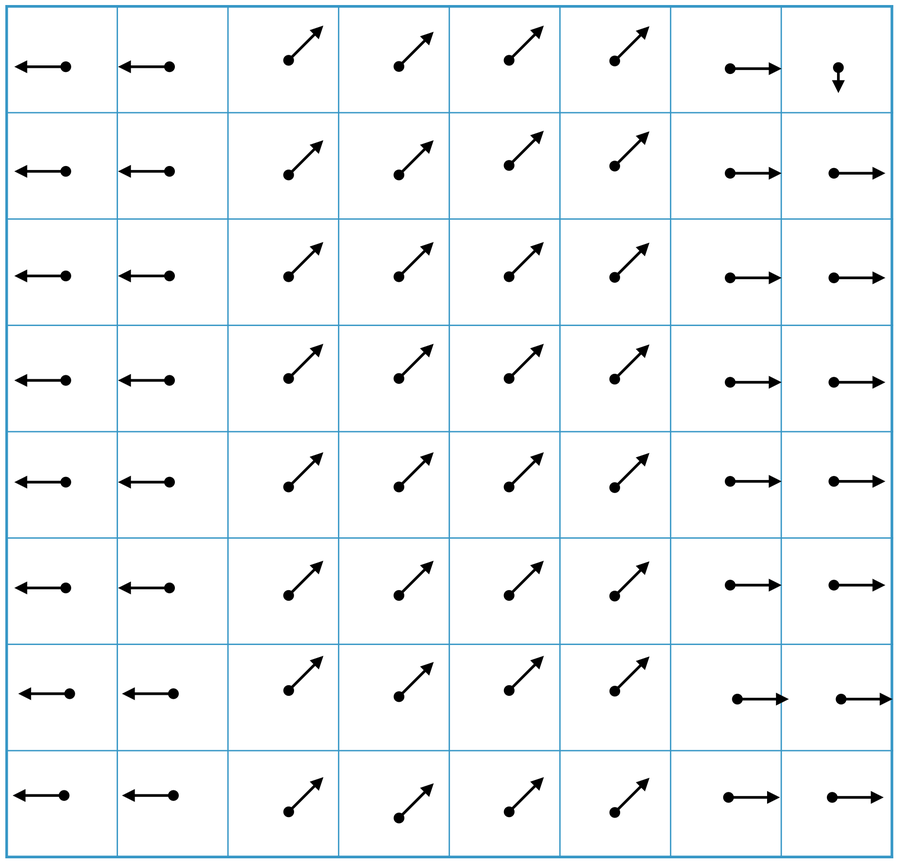
Gradient **Orientation Histogram**



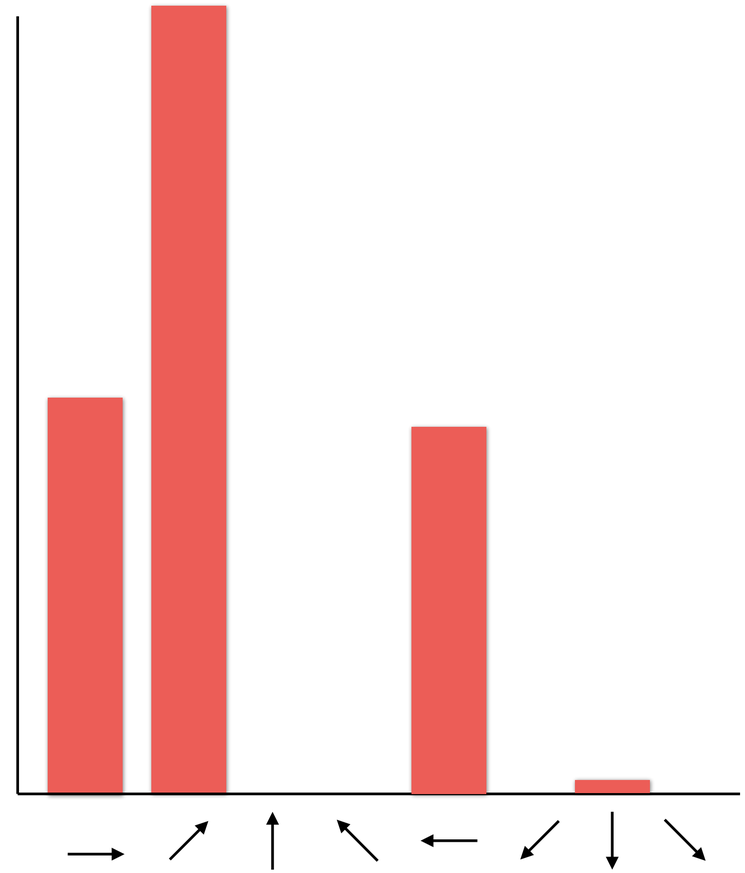


Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)

Gradient **Orientation Histogram**

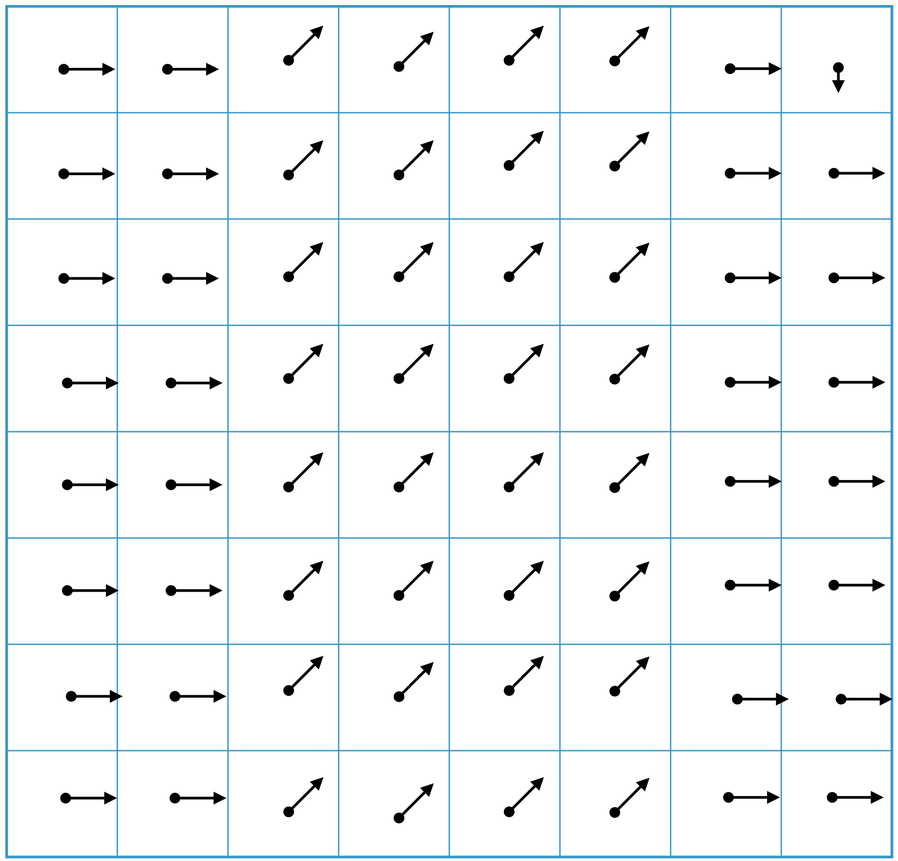


Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)

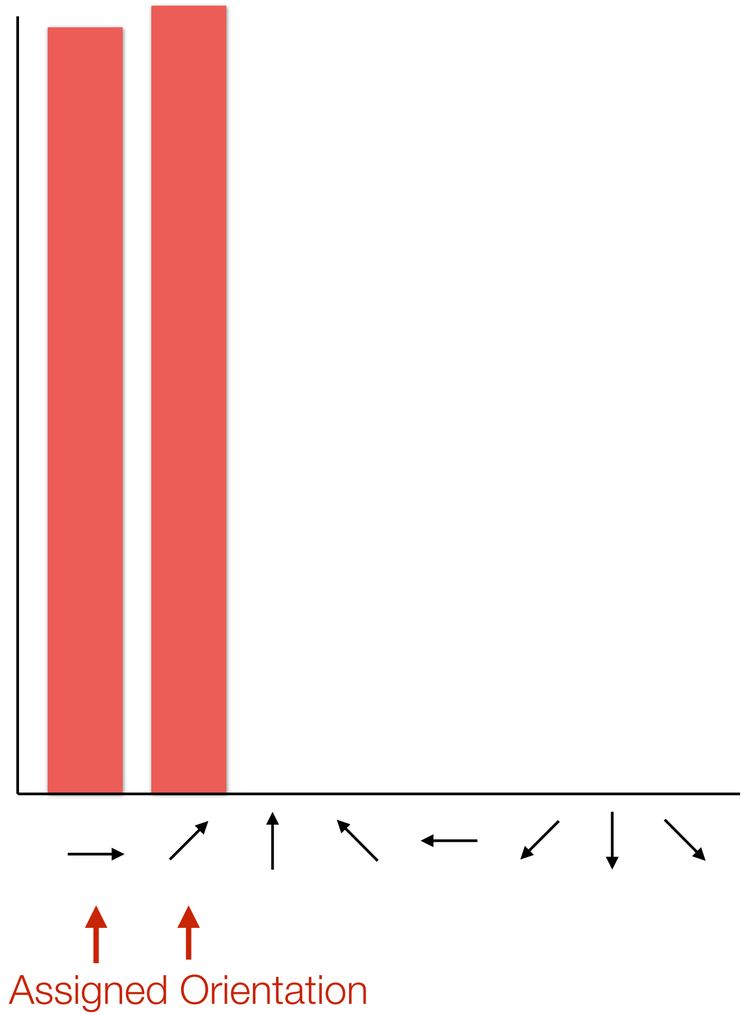


Assigned Orientation

Gradient **Orientation Histogram**

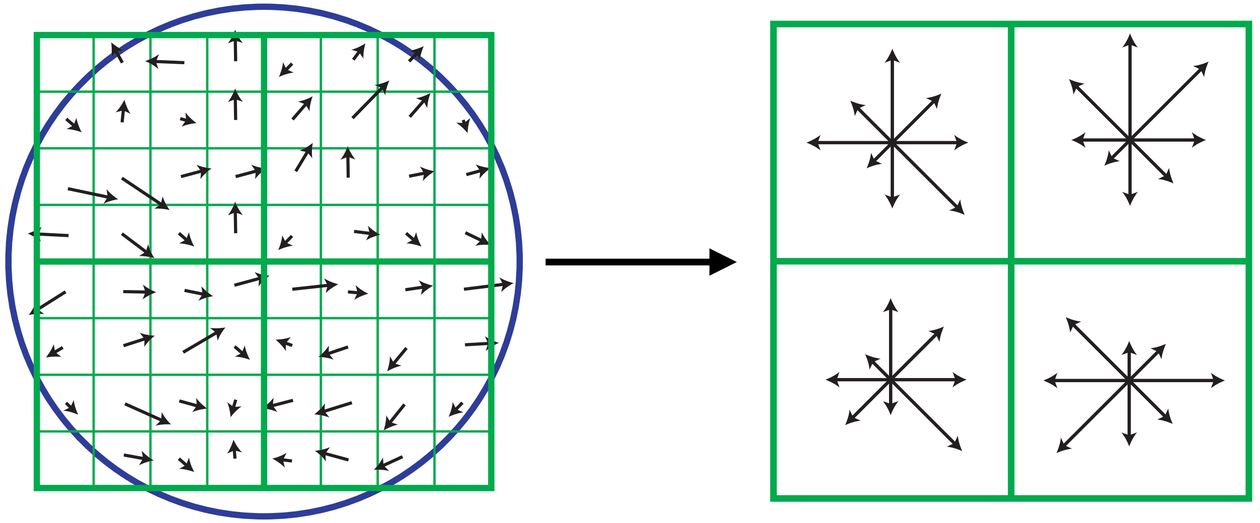


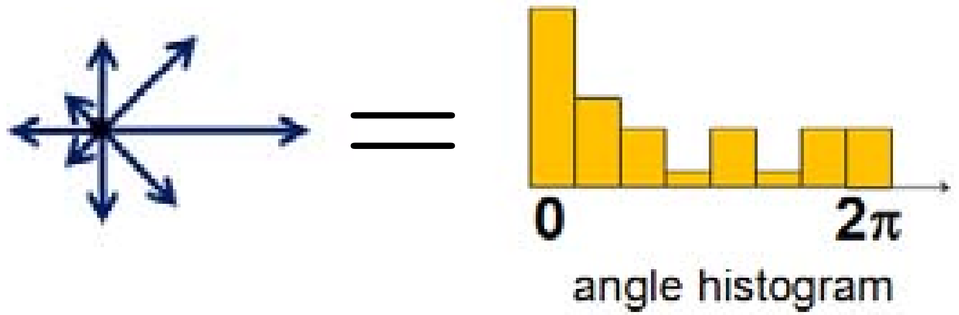
Arrows illustrate **gradient orientation** (direction)
and **gradient magnitude** (arrow length)



SIFT Descriptor

- Describe local region by distribution (over angle) of gradients





Each descriptor: 4 x 4 grid x 8 orientations = 128 dimensions

Photometric Invariance

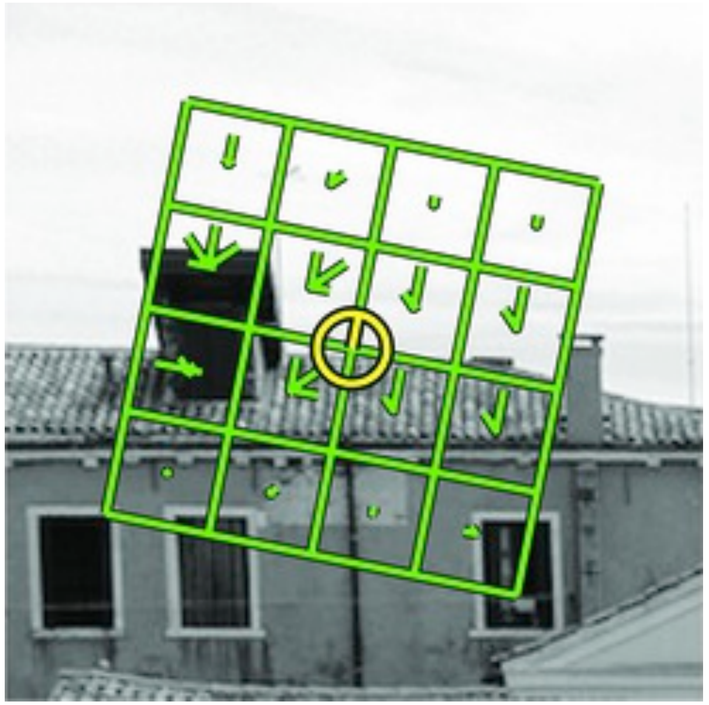
Descriptor is **normalized** to unit length (i.e. magnitude of 1) to reduce the effects of illumination change

— if brightness values are **scaled** (**multiplied**) by a constant, the gradients are scaled by the same constant, and the normalization cancels the change

— if brightness values are **increased/decreased** (**additive**) by a constant, the gradients do not change

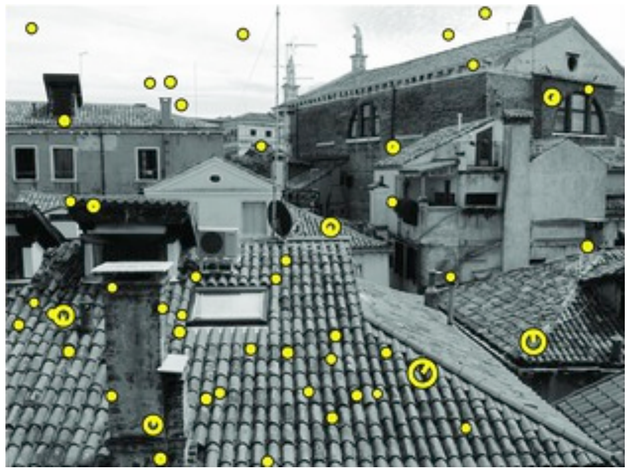
SIFT Recap

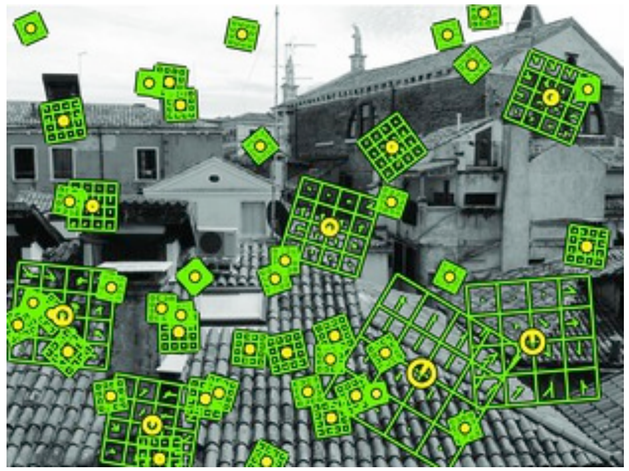
- **Detector**: find points that are maxima in a DOG pyramid
- Compute local orientation from gradient histogram
- This establishes a local coordinate frame with scale/orientation
- **Descriptor**: Build histograms over gradient orientations (8 orientations, 4x4 grid)
- Normalise the final descriptor to reduce the effects of illumination change

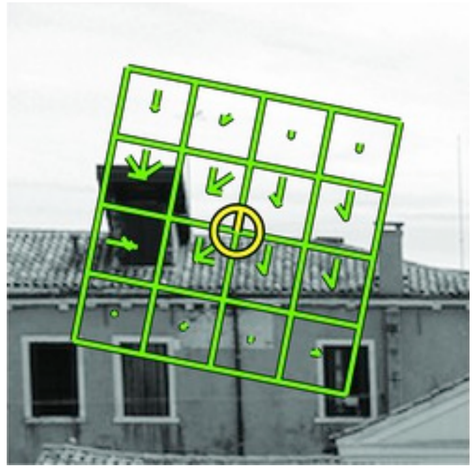


SIFT Matching

- Extract SIFT features from an image





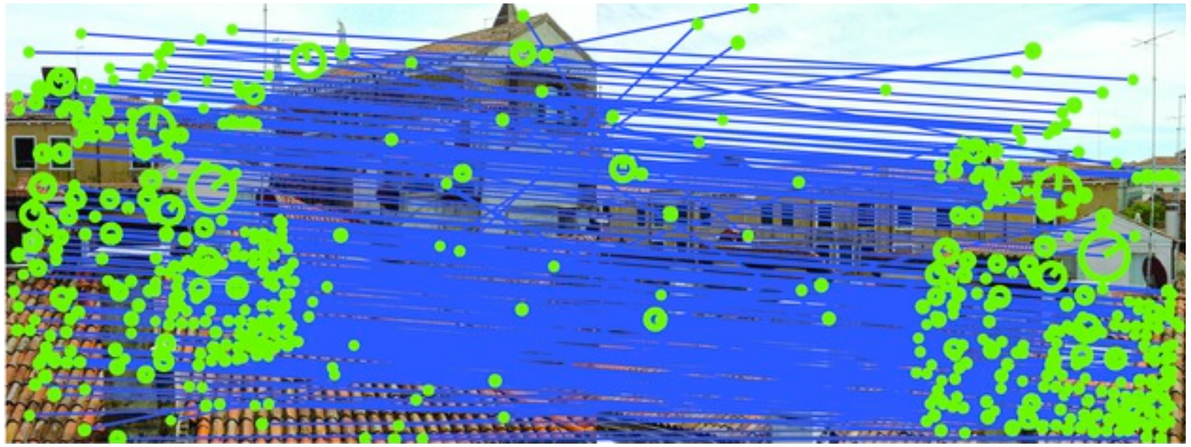


Each image might generate 100's or 1000's of SIFT descriptors

SIFT Matching

- Goal: Find all correspondences between a pair of images

- Extract and match all SIFT descriptors from both images



SIFT Matching

- Each SIFT feature is represented by 128 numbers
- Feature matching becomes task of finding a nearby 128-d vector
- Nearest-neighbour matching:

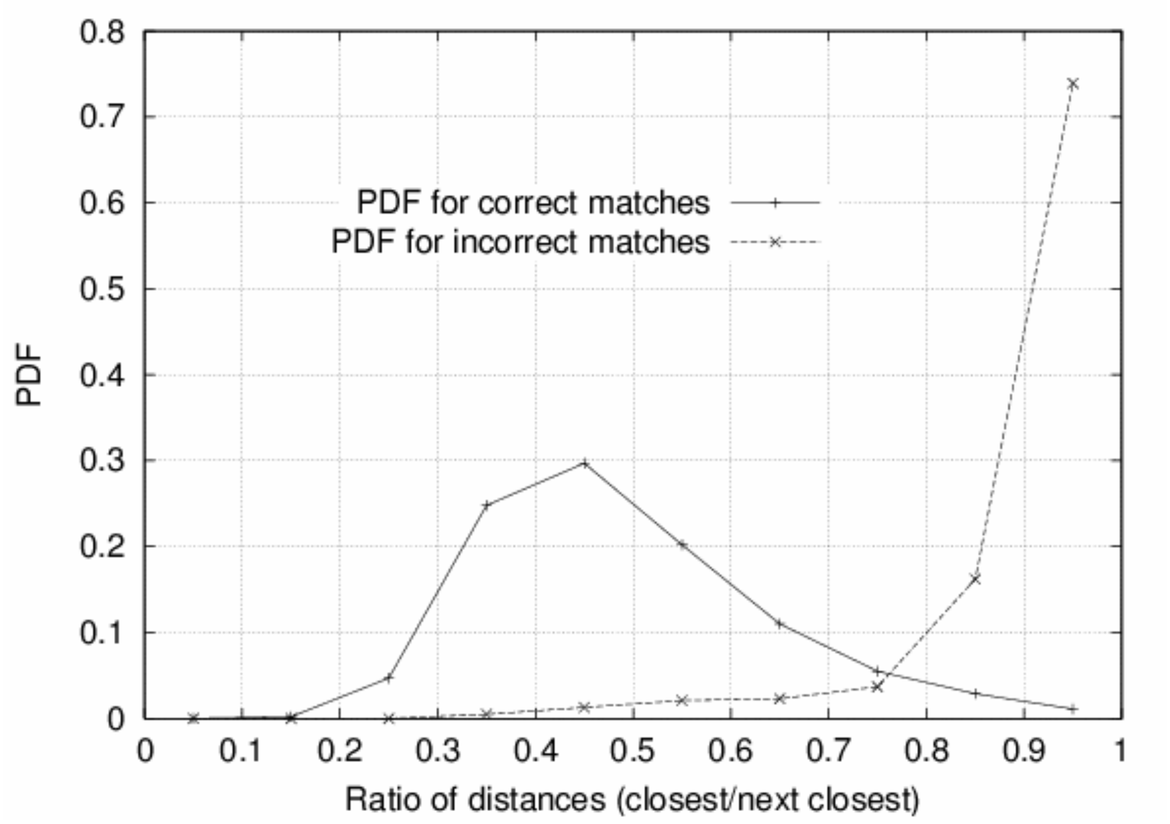
$$NN(j) = \arg\min_i |\mathbf{x}_i - \mathbf{x}_j|,\ i \neq j$$

- Linear time, but good approximation algorithms exist
- e.g., Best Bin First K-d Tree [Beis Lowe 1997], FLANN (Fast Library for Approximate Nearest Neighbours) [Muja Lowe 2009]

Match **Ratio Test**

Compare ratio of distance of **nearest** neighbour (1NN) to **second** nearest (2NN) neighbour — this will be a non-matching point

Rule of thumb: d(1NN) < 0.8 * d(2NN) for good match

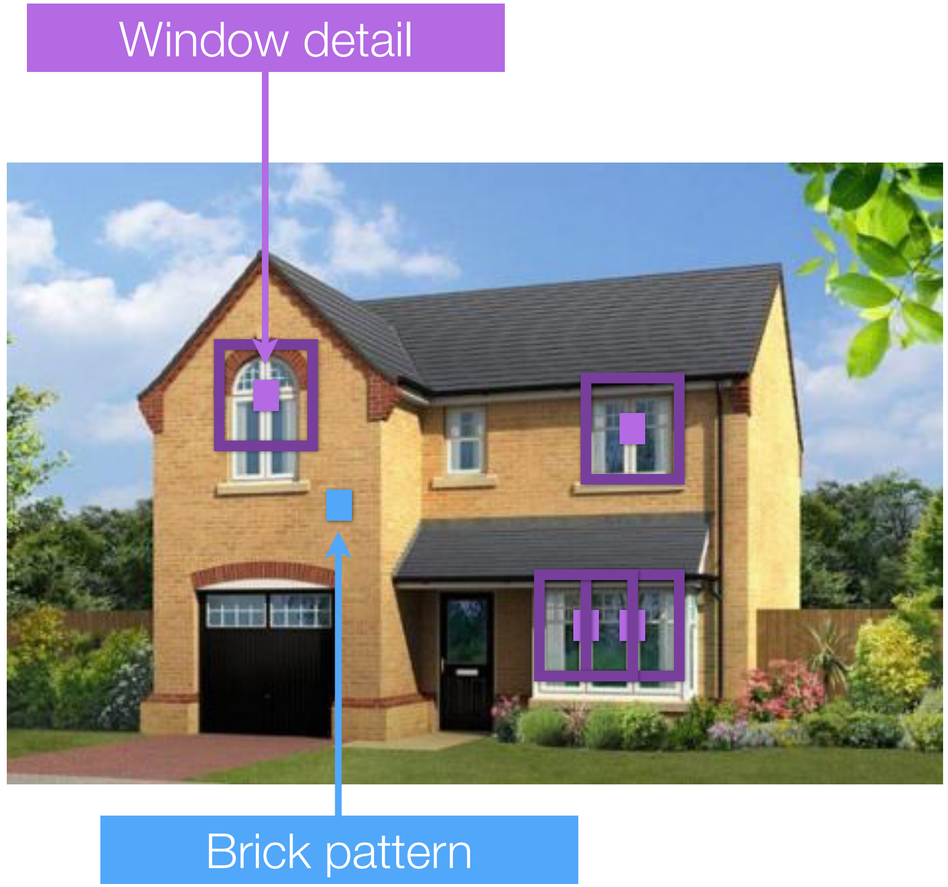


Failure Case: **Repetitive** Structures

Repetitive structures cause problems for feature matching

Multiple locations in an image provide good matches and have similar matching scores

They are particularly common in man-made environments

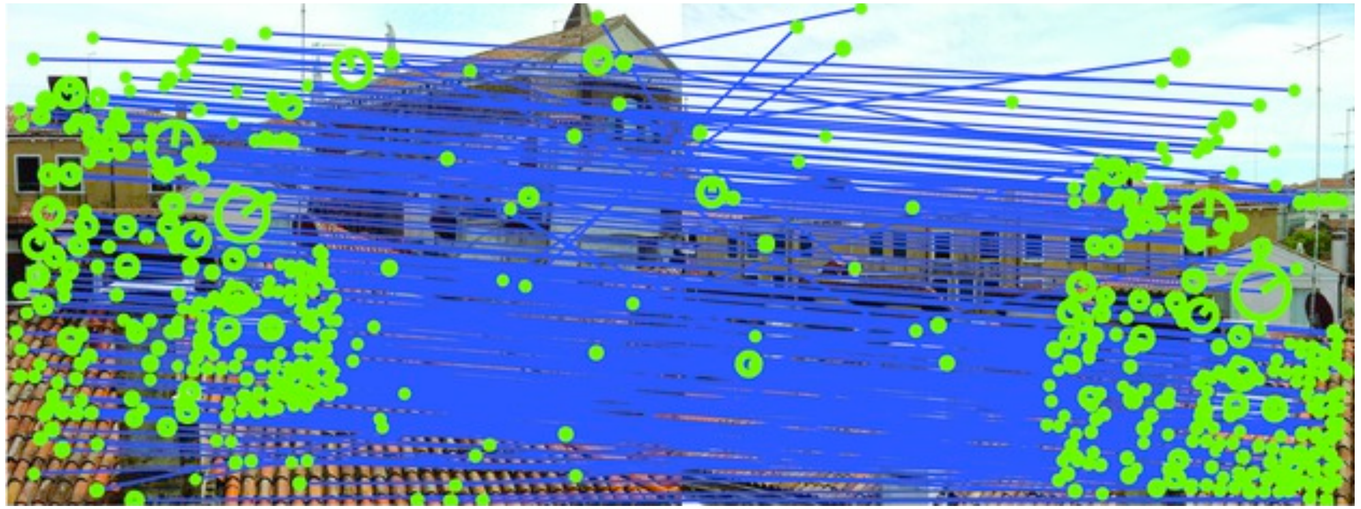


Assignment 4

- Try out the **Nearest Neighbour Matching** section in Assignment 4

SIFT Matching

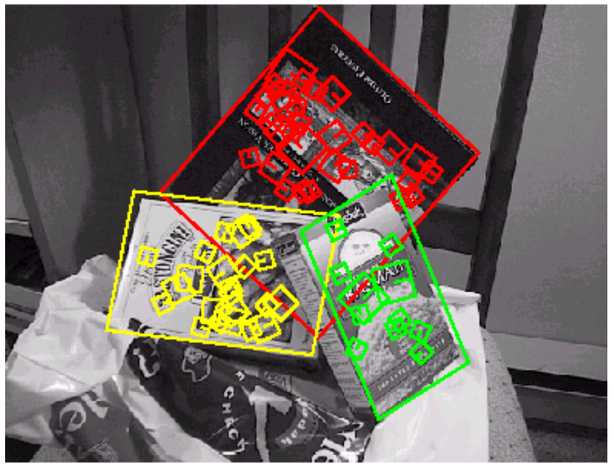
- Feature matching returns a set of noisy correspondences
- To get further, we will have to know something about the **geometry** of the images



Planar Object **Instance Recognition**

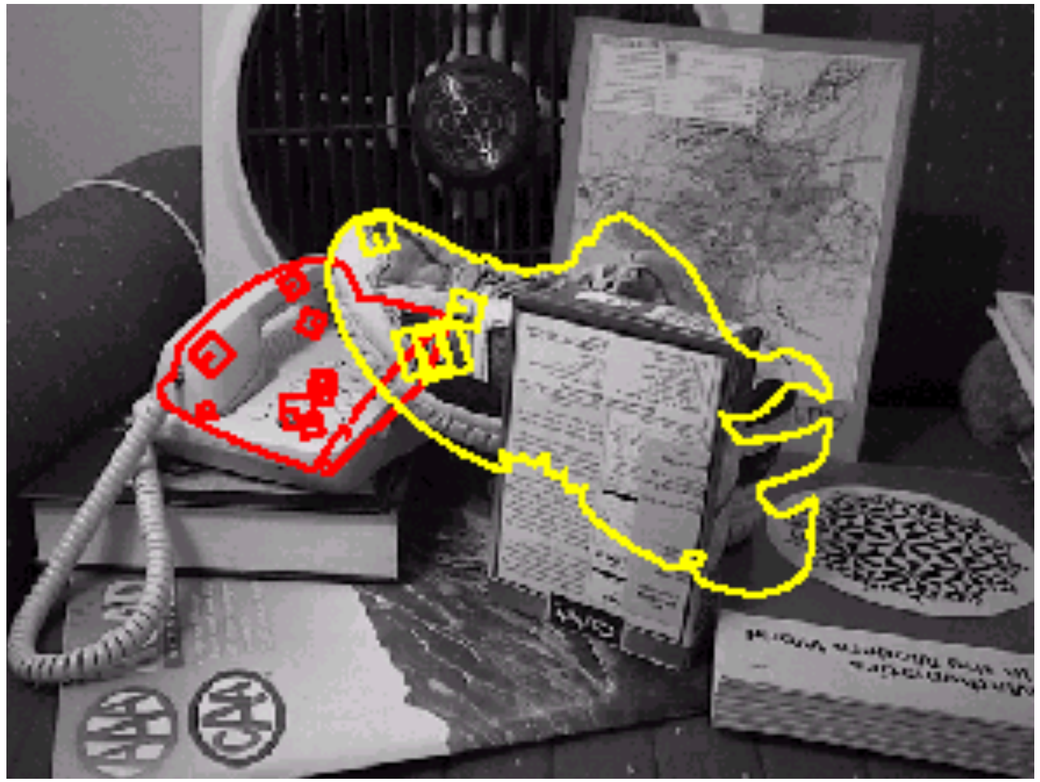
Database of planar objects

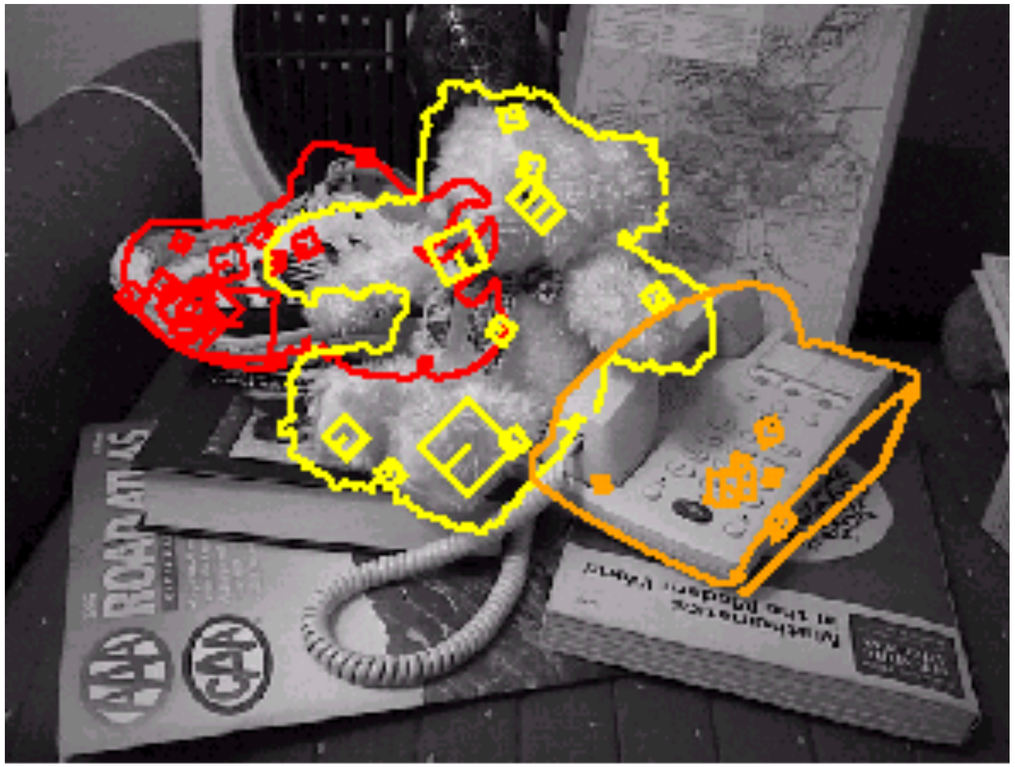
Instance recognition



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Recognition under **Occlusion**

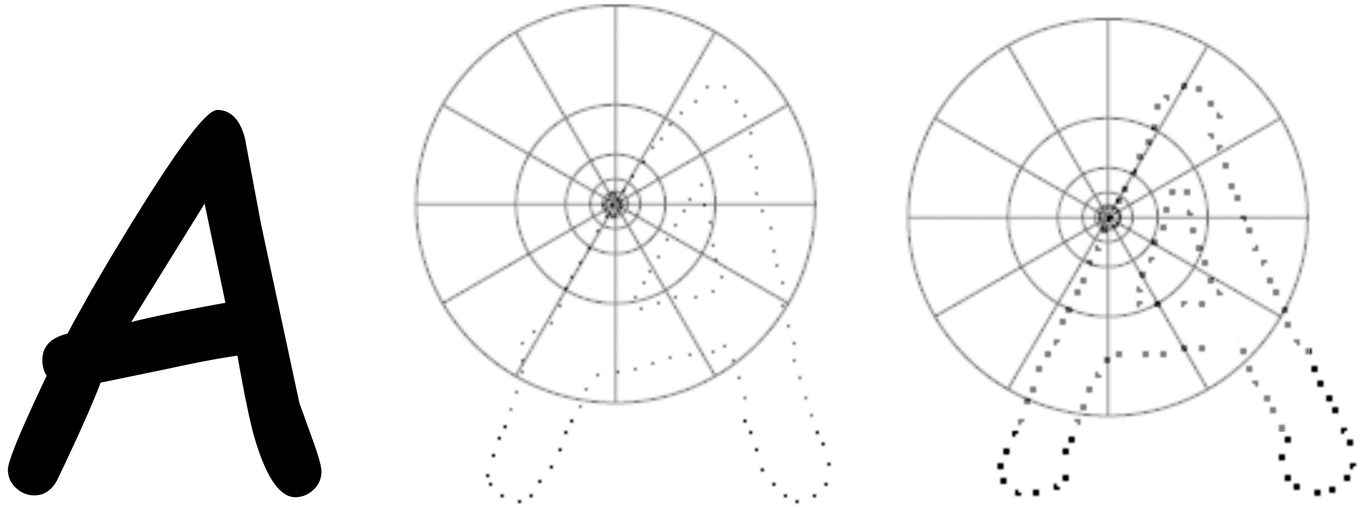




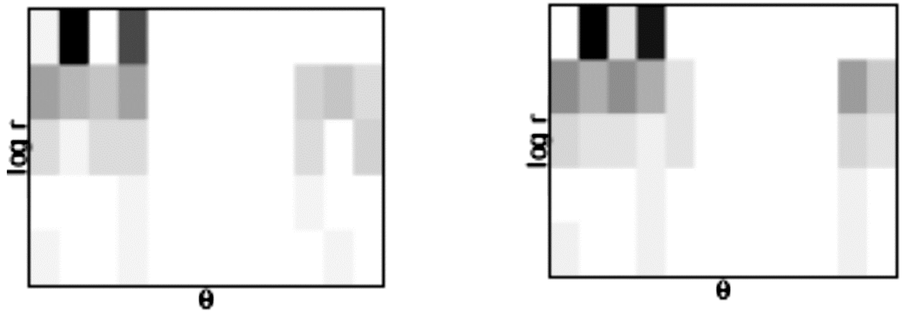
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Shape Context

- Useful for matching with contours



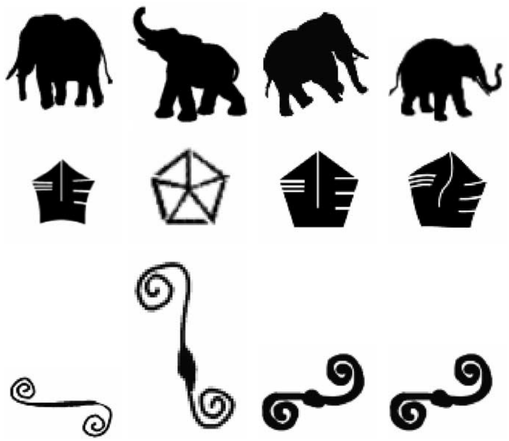
Descriptor is log polar histogram



[Belongie Malik 2000]

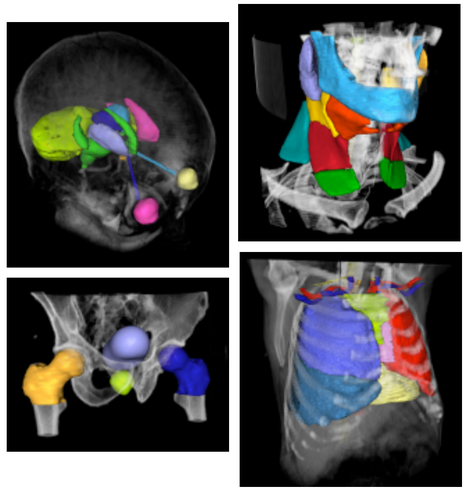
Choosing Features

- The best choice of features is usually application dependent



Shape context?

SIFT?



Something else?

Learning Descriptors

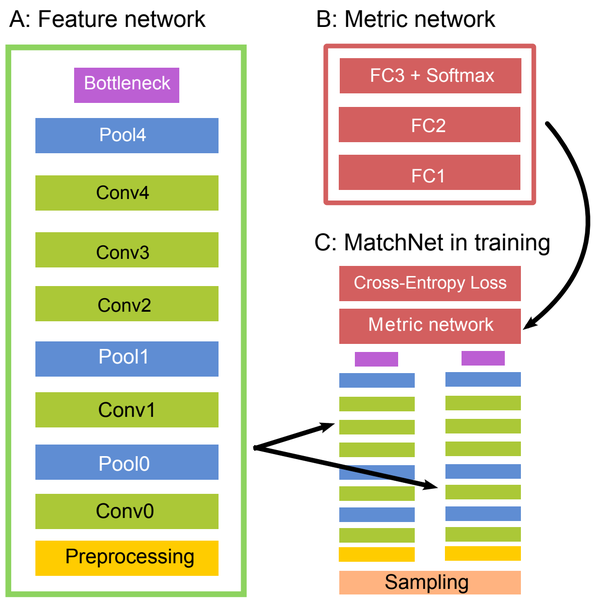
- Descriptor design as a learning (embedding) problem

[Winder Brown 2007]

Learning Descriptors

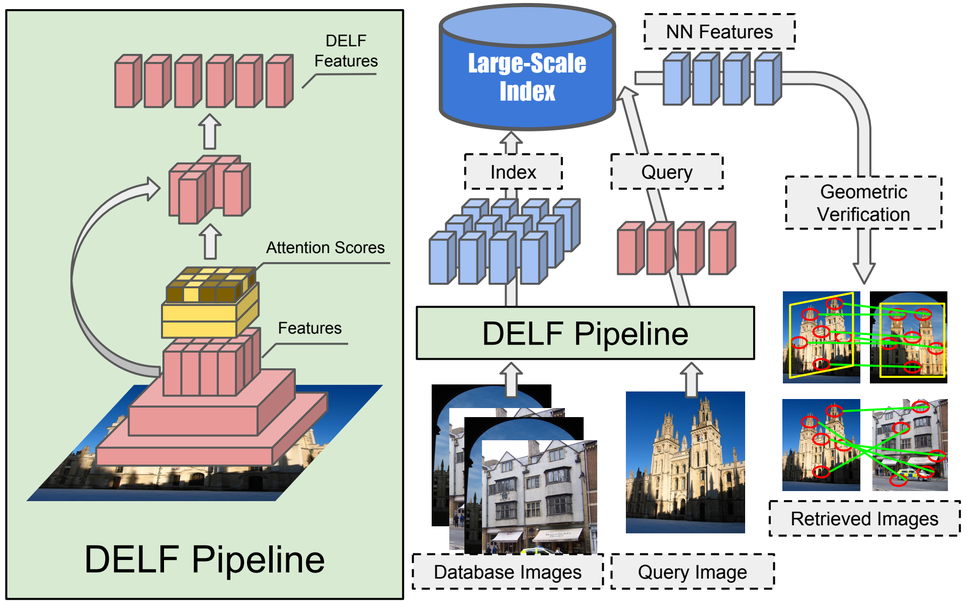
- Deep networks for descriptor learning

Patch labels



[MatchNet
Han et al 2015]

Image labels, also learns interest function



[DELF
Noh et al 2017]

Menu for Today

Topics:

- **Correspondence** Problem
- **Patch** matching
- **Invariance**, geometric, photometric
- **SIFT** = Scale Invariant Feature Transform

Readings:

- **Today's** Lecture: Szeliski Chapter 7, Forsyth & Ponce 5.4

Reminders:

- **Assignment 4**: RANSAC and Panorama Stitching — **now available**