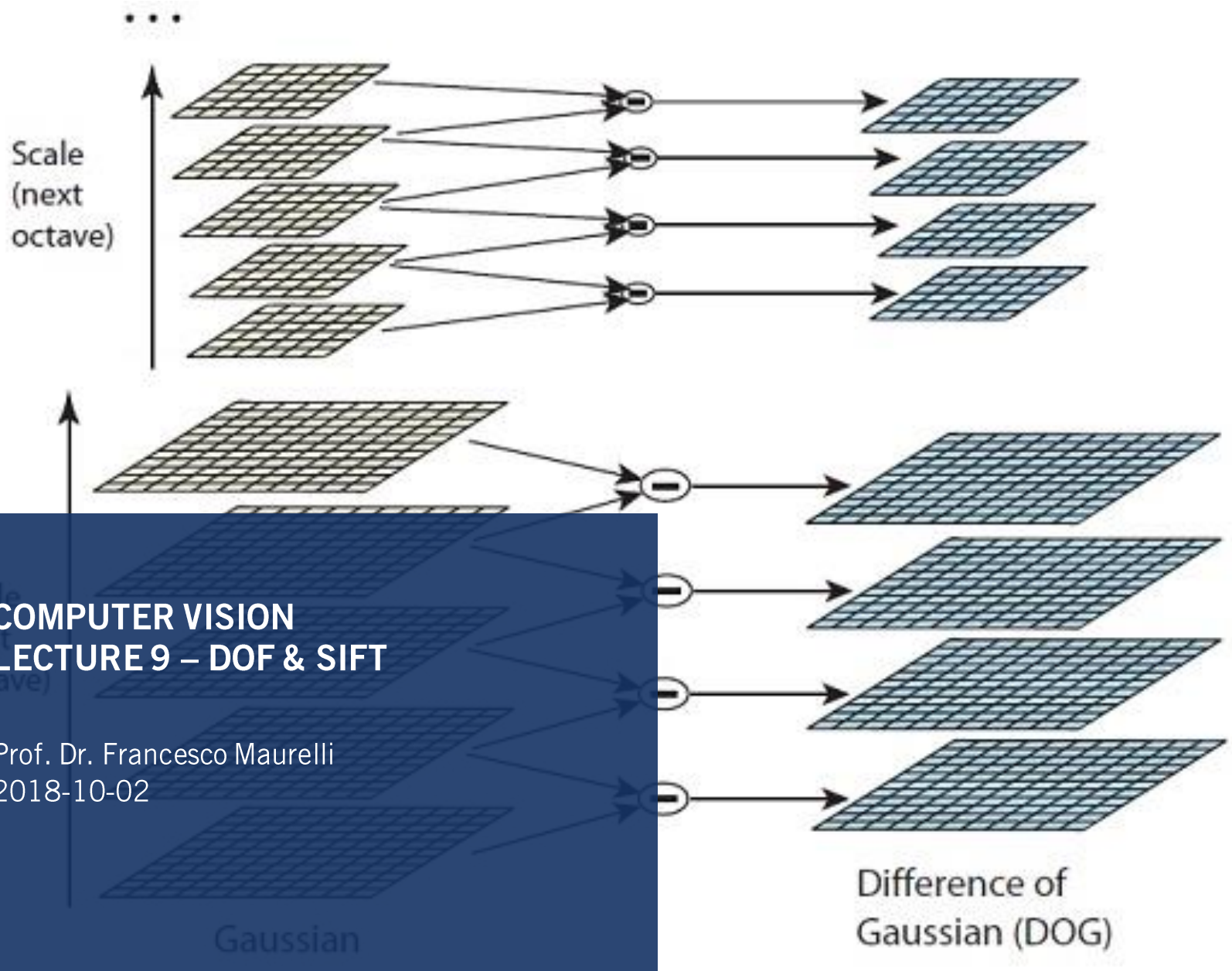
JACOBS UNIVERSITY



COMPUTER VISION
LECTURE 9 – DOF & SIFT

Prof. Dr. Francesco Maurelli
2018-10-02

WHAT WE WILL LEARN TODAY?

- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HOG: another image descriptor
- Application: Panorama

JACOBS UNIVERSITY

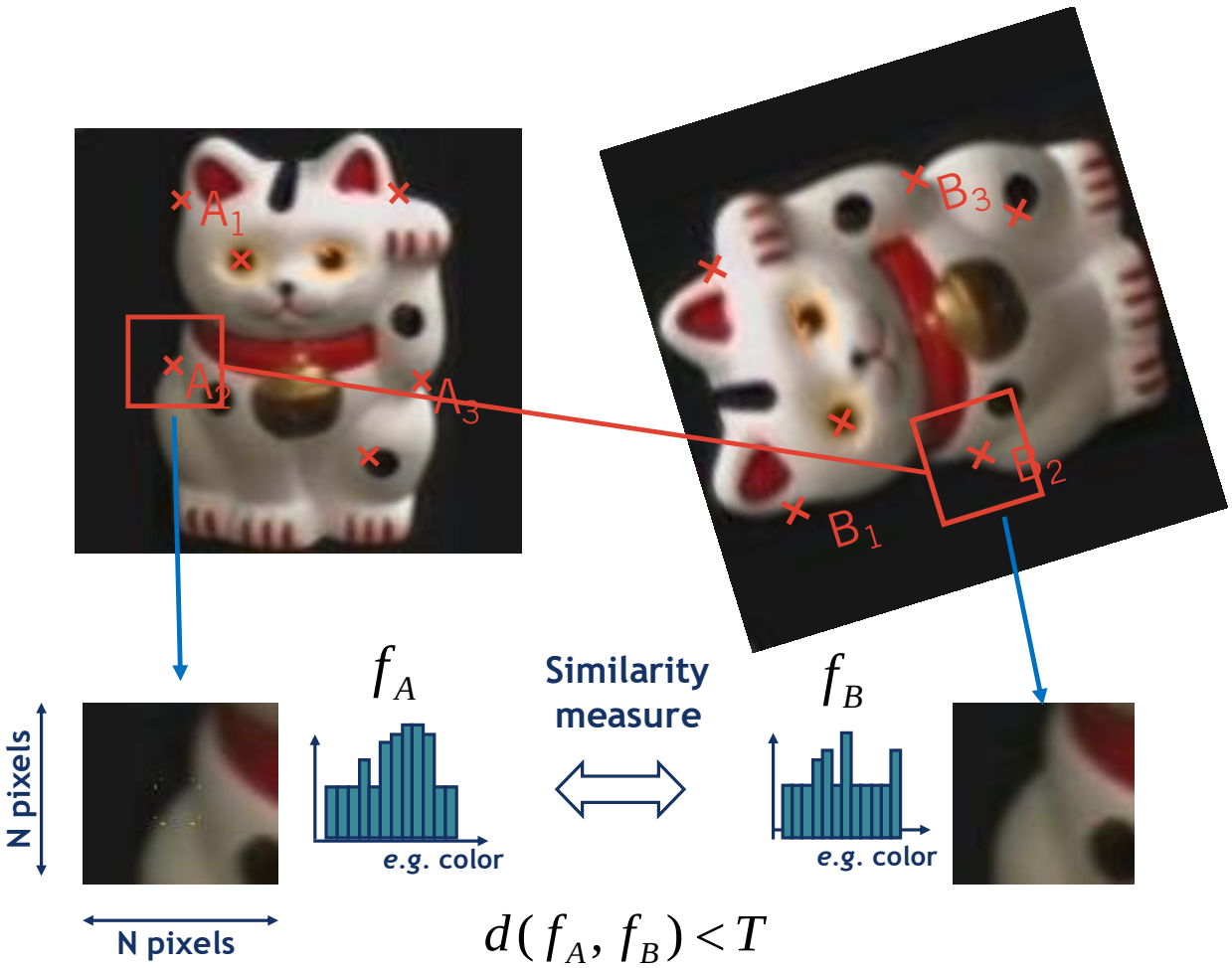
Local invariant features

- Motivation
- Requirements, invariances

Keypoint localization

- Harris corner detector

GENERAL APPROACH



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

Slide credit: Bastian Leibe

JACOBS UNIVERSITY

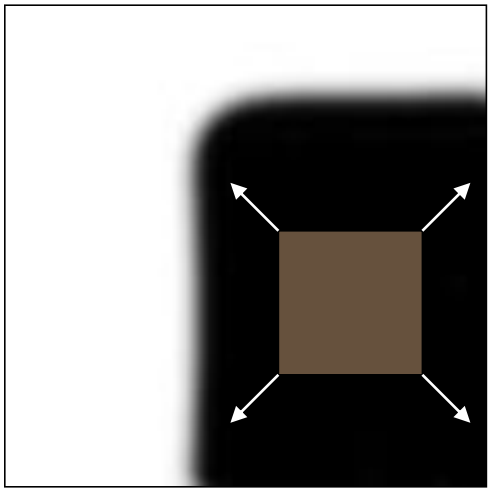
Local invariant features

- Motivation
- Requirements, invariances

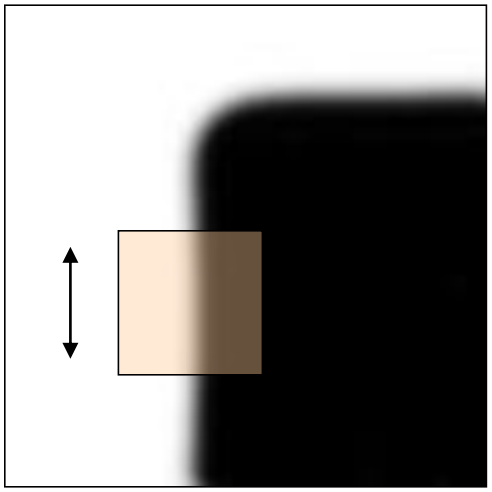
Keypoint localization

- Harris corner detector

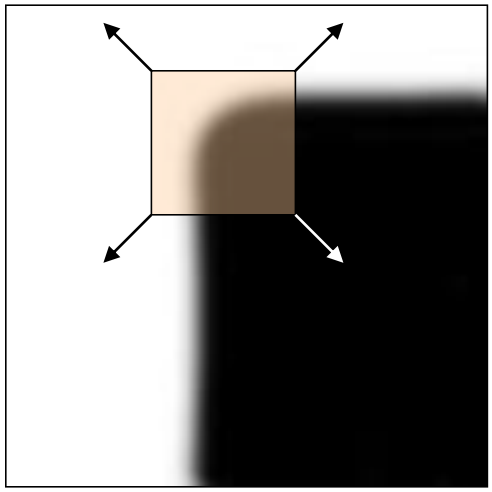
QUICK REVIEW: HARRIS CORNER DETECTOR



"flat" region:
no change in all directions



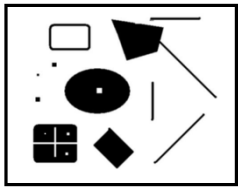
"edge":
no change along the edge direction



"corner":
significant change in all directions

Slide credit: Alyosha Efros

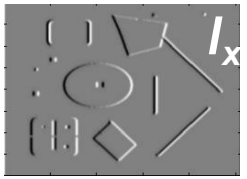
JACOBS UNIVERSITY

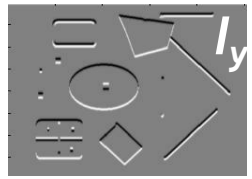


Compute second moment matrix (autocorrelation matrix)

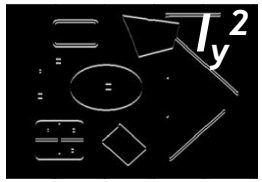
$$M(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

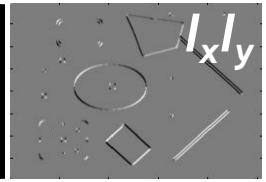
1. **Image derivatives**



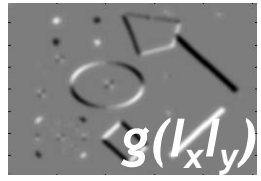


2. **Square of derivatives**





3. **Gaussian filter $g(\sigma_I)$**



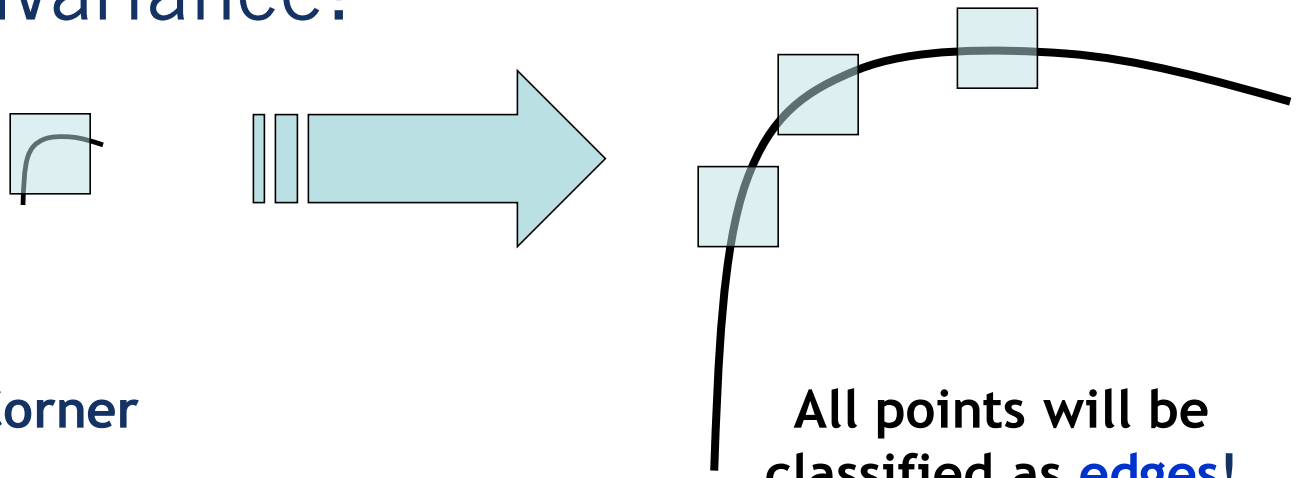
4. **Cornerness function - two strong eigenvalues**

$$q = \det[M(s_I, s_D)] - a[\text{trace}(M(s_I, s_D))]^2$$
$$= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. **Perform non-maximum suppression**

SUMMARY: HARRIS DETECTOR

- Translation invariance
- Rotation invariance
- Scale invariance?



Corner

All points will be classified as edges!

Not invariant to image scale!

Slide credit: Kristen Grauman

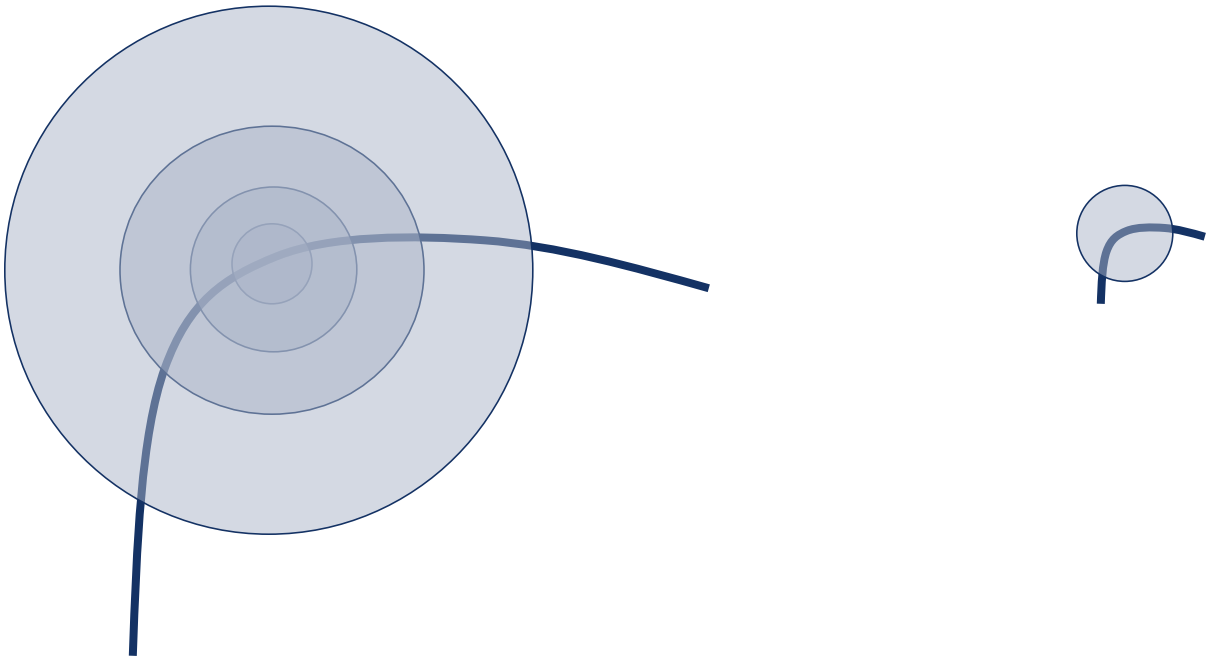
WHAT WE WILL LEARN TODAY?

- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HOG: another image descriptor
- Application: Panorama

DOES SCALE MATTER?

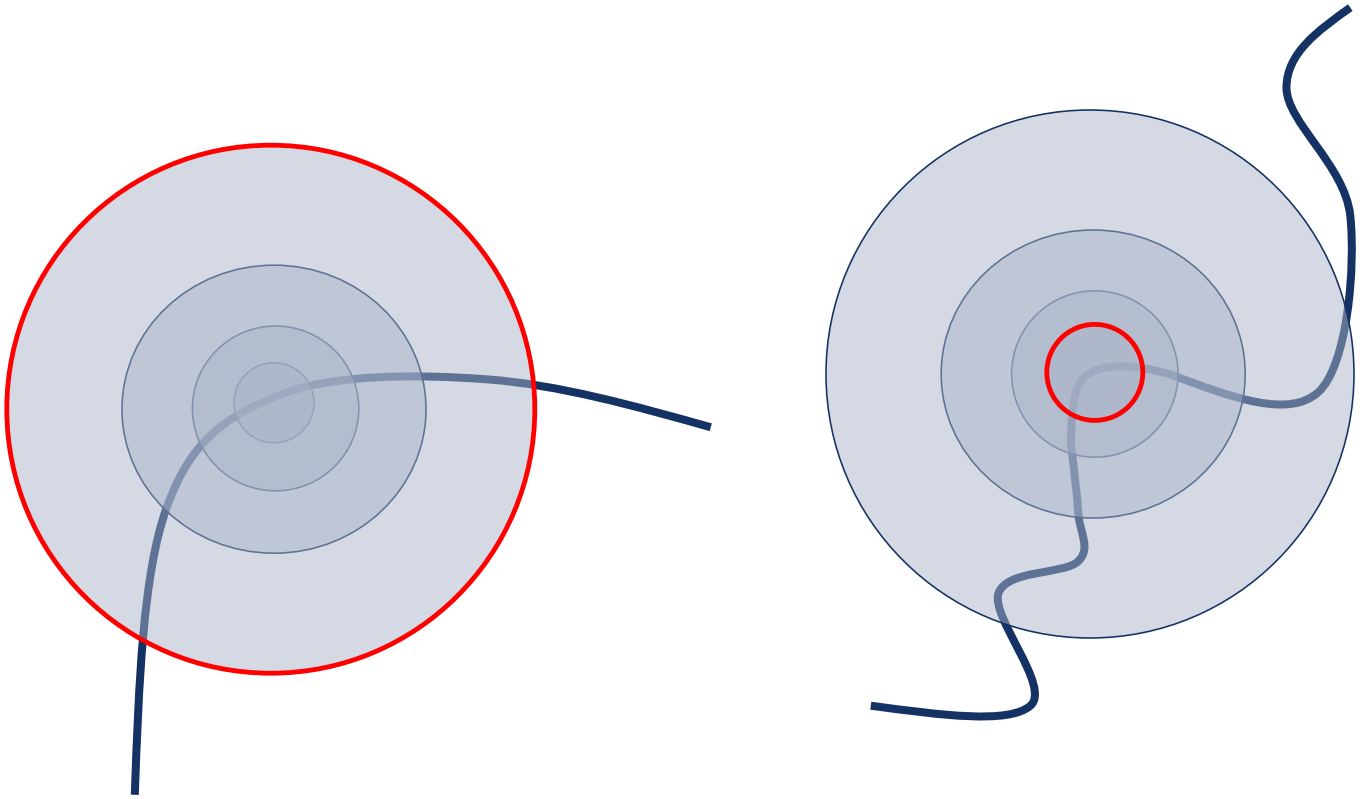
1. When detecting corners, the `scale` of the window you use can change the corners you detect.

1. Consider regions (e.g. circles) of different sizes around a point
2. Find regions of corresponding sizes that will look the same in both images?



JACOBS UNIVERSITY

1. The problem: how do we choose corresponding circles ***independently*** in each image?

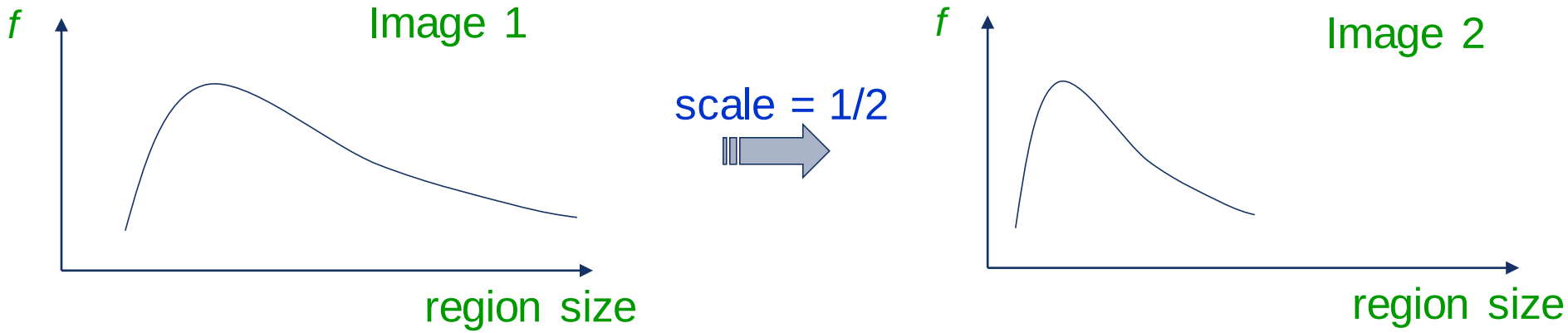


JACOBS UNIVERSITY

1. Solution:
   - Design a function on the region (circle), which is "scale invariant" (the same for corresponding regions, even if they are at different scales)

     Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

     – For a point in one image, we can consider it as a function of region size (circle radius)



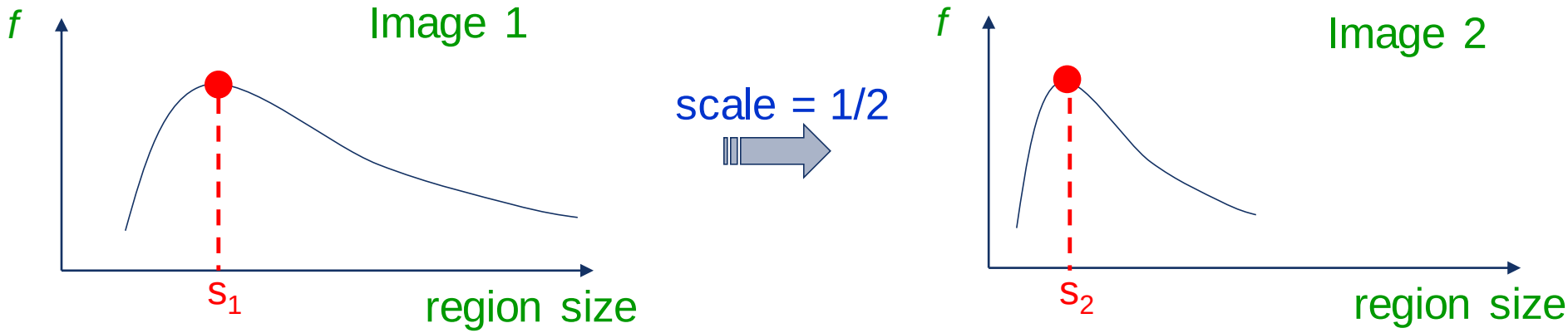
JACOBS UNIVERSITY

1. Common approach:

   Take a local maximum of this function

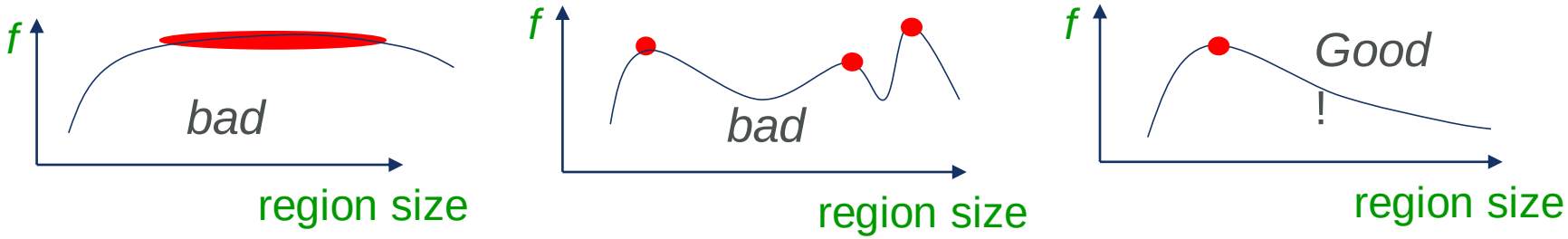
- Observation: region size, for which the maximum is achieved, should be *co-variant* with image scale.

Important: this scale invariant region size is found in each image independently!



JACOBS UNIVERSITY

A "good" function for scale detection: has one stable sharp peak



- For usual images: a good function would be one which responds to contrast (sharp local intensity change)

JACOBS UNIVERSITY

Functions for determining scale

$$f = \text{Kernel} * \text{Image}$$

Kernels:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

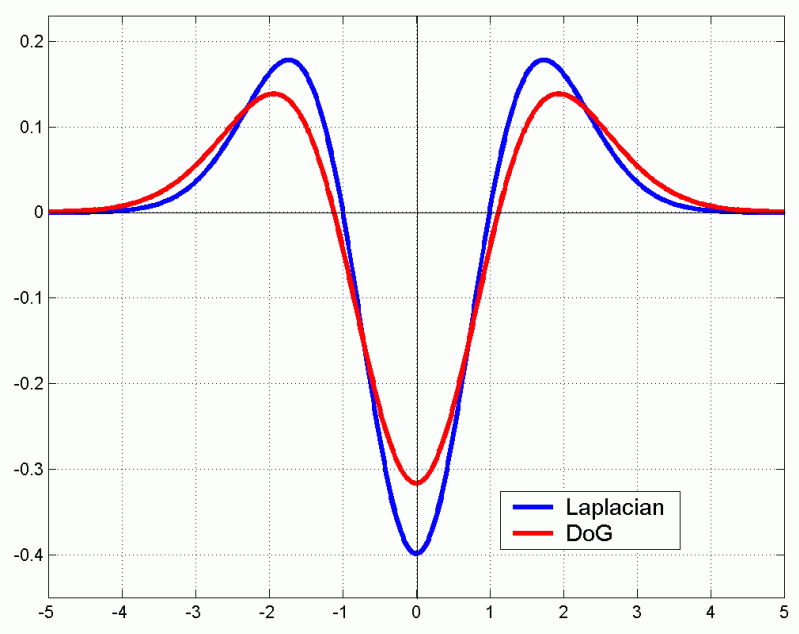
(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

where Gaussian

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



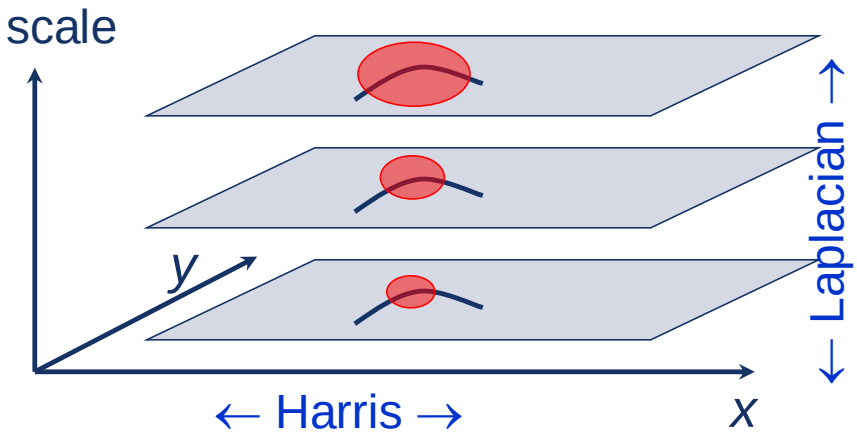
Note: both kernels are invariant to *scale* and *rotation*

JACOBS UNIVERSITY

Harris-Laplacian[1]

Find local maximum of:

- Harris corner detector in space (image coordinates)
- Laplacian in scale



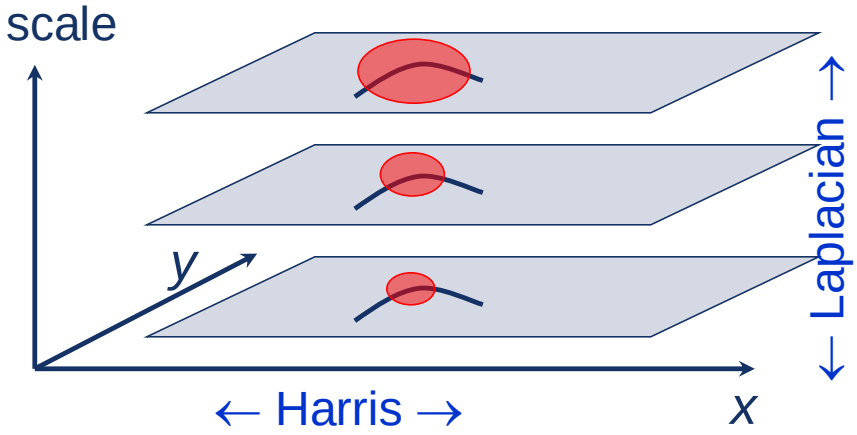
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

JACOBS UNIVERSITY

Harris-Laplacian[1]

Find local maximum of:

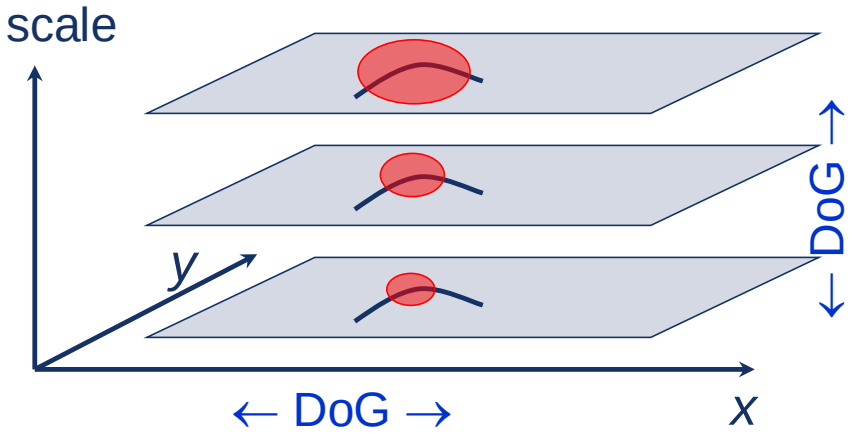
- Harris corner detector in space (image coordinates)
- Laplacian in scale



SIFT (Lowe)[2]

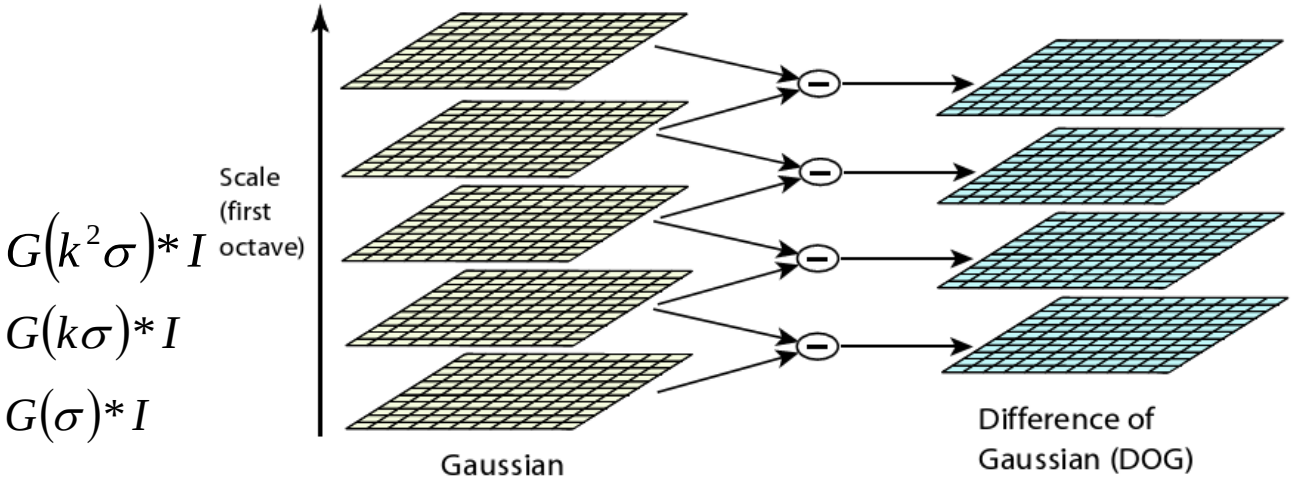
Find local maximum of:

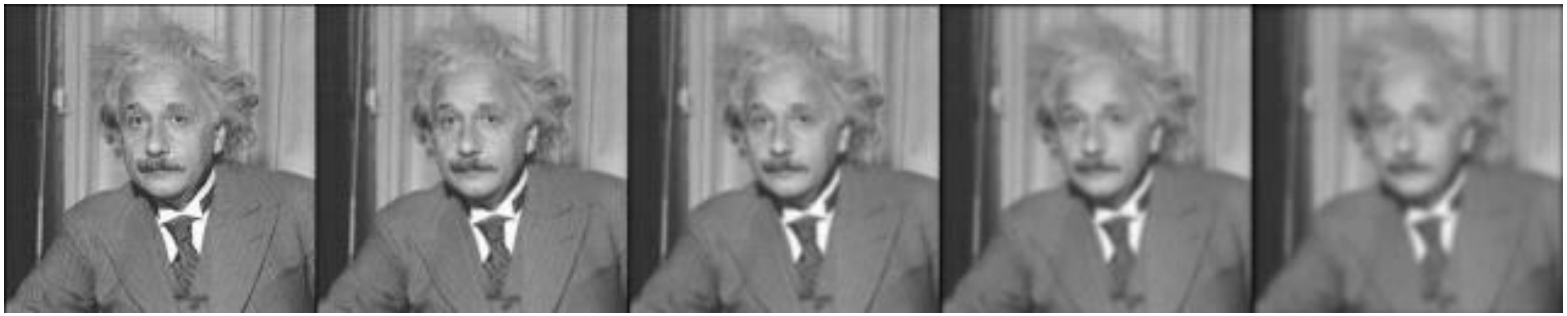
- Difference of Gaussians in space and scale



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

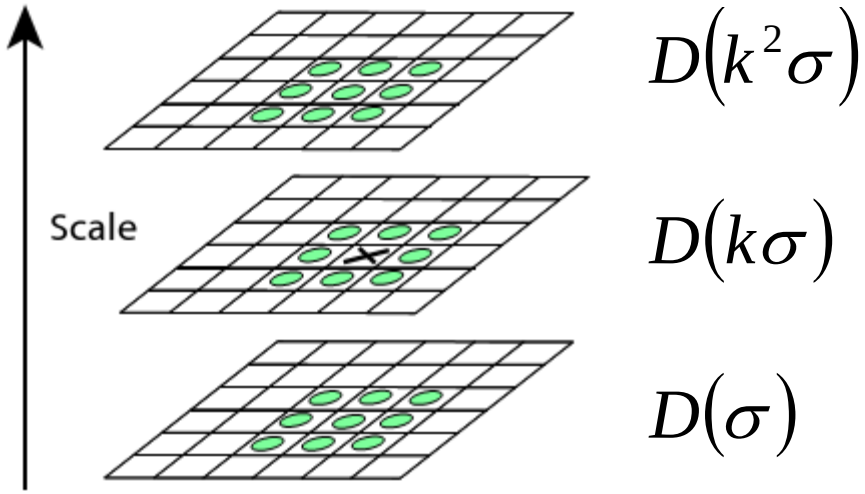
DIFFERENCE-OF-GAUSSIANS





JACOBS UNIVERSITY

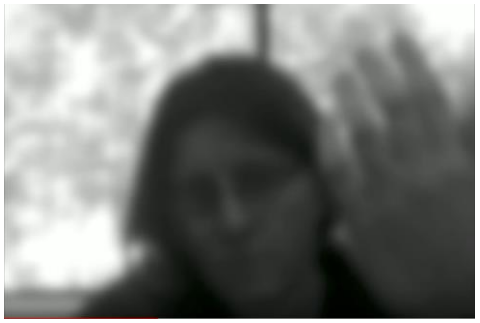
Choose all extrema within 3x3x3 neighborhood.



X is selected if it is larger or smaller than all 26 neighbors

EXAMPLE OF GAUSSIAN KERNEL

Original video

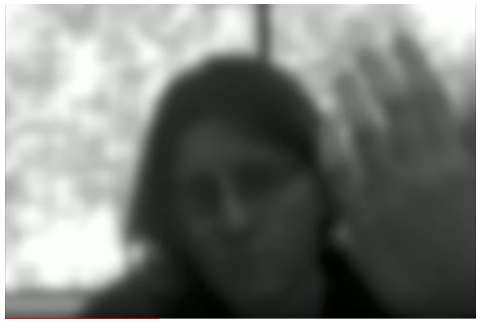


Blurred with a gaussian kernel

Blurred with a different gaussian kernel

EXAMPLE OF GAUSSIAN KERNEL

Original video



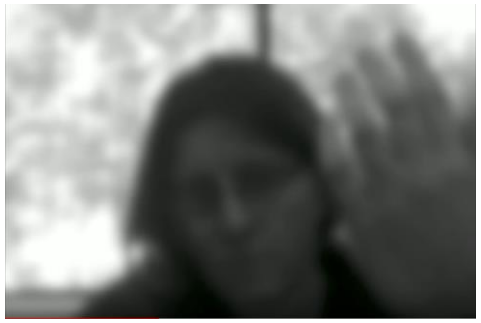
Blurred with a gaussian kernel

Blurred with a different gaussian kernel

What happens if you subtract one blurred image from another?

DIFFERENCE OF GAUSSIANS (DOG)

Original video



Blurred with a gaussian kernel: k_1

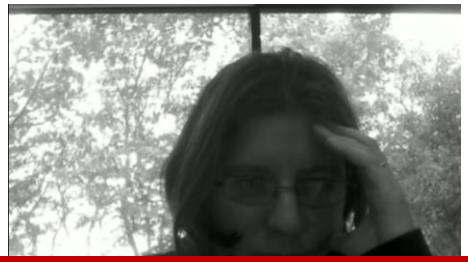
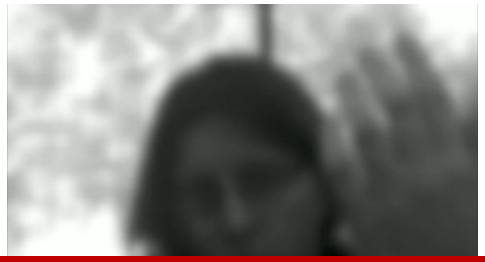
Blurred with a different gaussian kernel: k_2

DoG: $k_1 - k_2$

DoG: $k_1 - k_3$

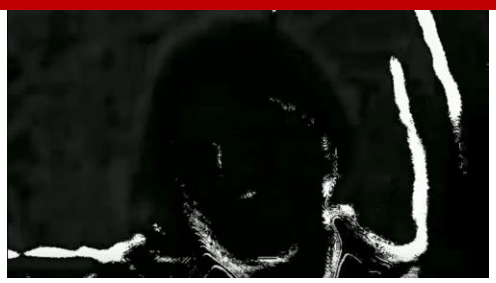
DoG: $k_1 - k_4$

DIFFERENCE OF GAUSSIANS (DOG)


At different resolutions of kernel size, we see different fine details of the image. In other words, we can capture keypoints at varying scales.





DoG: $k_1 - k_2$

DoG: $k_1 - k_3$

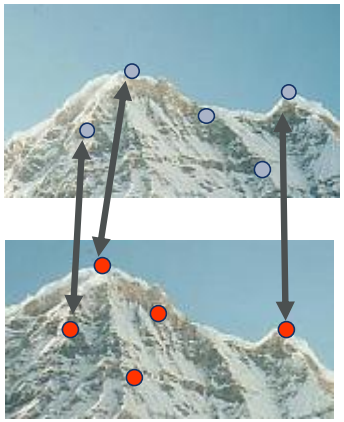
DoG: $k_1 - k_4$

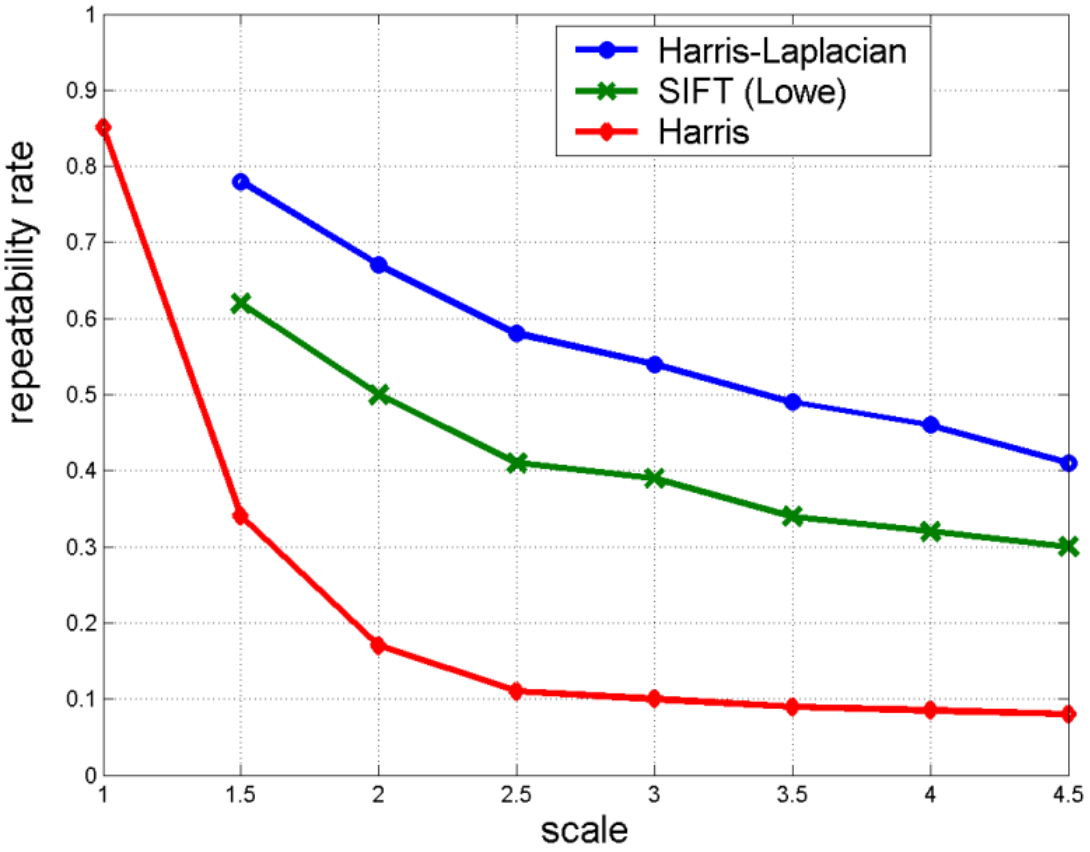
JACOBS UNIVERSITY

Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\#\text{ correspondences}}{\#\text{ possible correspondences}}$$





K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

JACOBS UNIVERSITY

1. Given: two images of the same scene with a large *scale difference* between them
2. Goal: find *the same* interest points *independently* in each image
3. Solution: search for *maxima* of suitable functions in *scale* and in *space* (over the image)

Methods:

1. Harris-Laplacian [Mikolajczyk, Schmid]: maximize Laplacian over scale, Harris' measure of corner response over the image
2. SIFT [Lowe]: maximize Difference of Gaussians over scale and space

JACOBS UNIVERSITY

So we have can detect keypoints at varying scales. But what can we do with those keypoints?

Things we would like to do:

- Search:
  We would need to find similar key points in other images
- Panorama
  Match keypoints from one image to another.
- Etc…

For all such applications, we need a way of `describing` the keypoints.

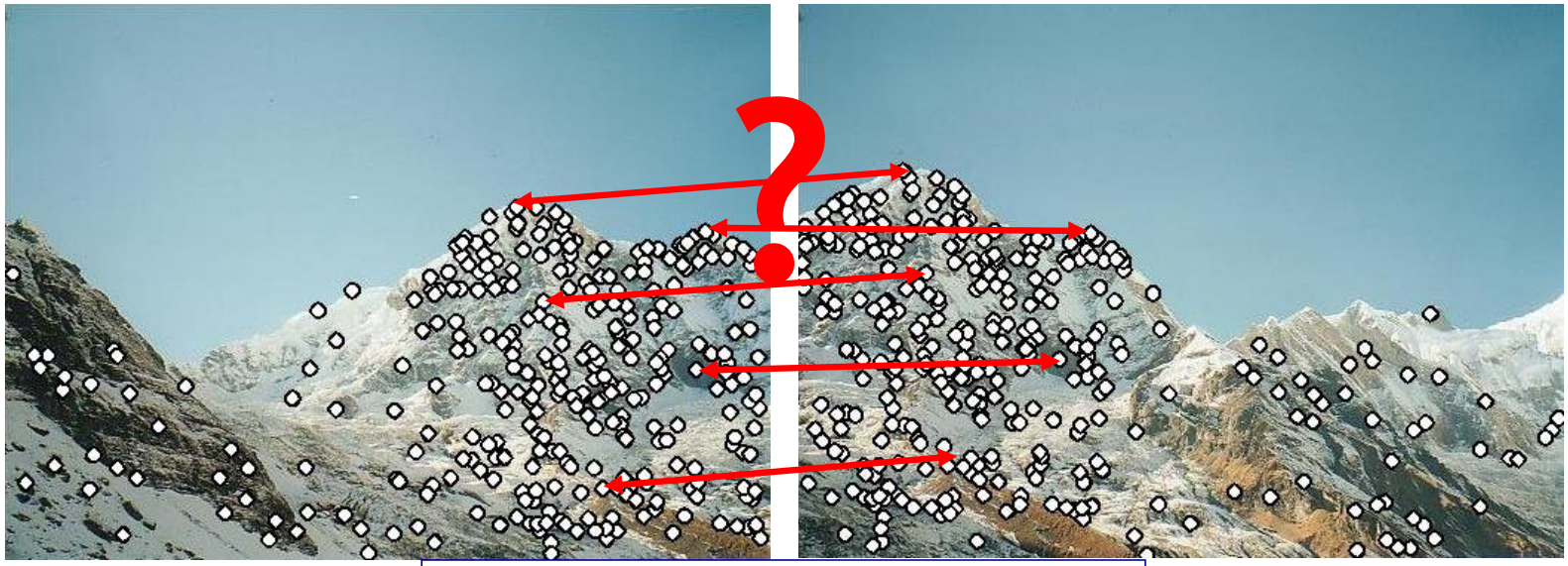
WHAT WE WILL LEARN TODAY?

- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HOG: another image descriptor
- Application: Panorama

JACOBS UNIVERSITY

1. We know how to detect points
2. Next question:

How to *describe* them for matching?



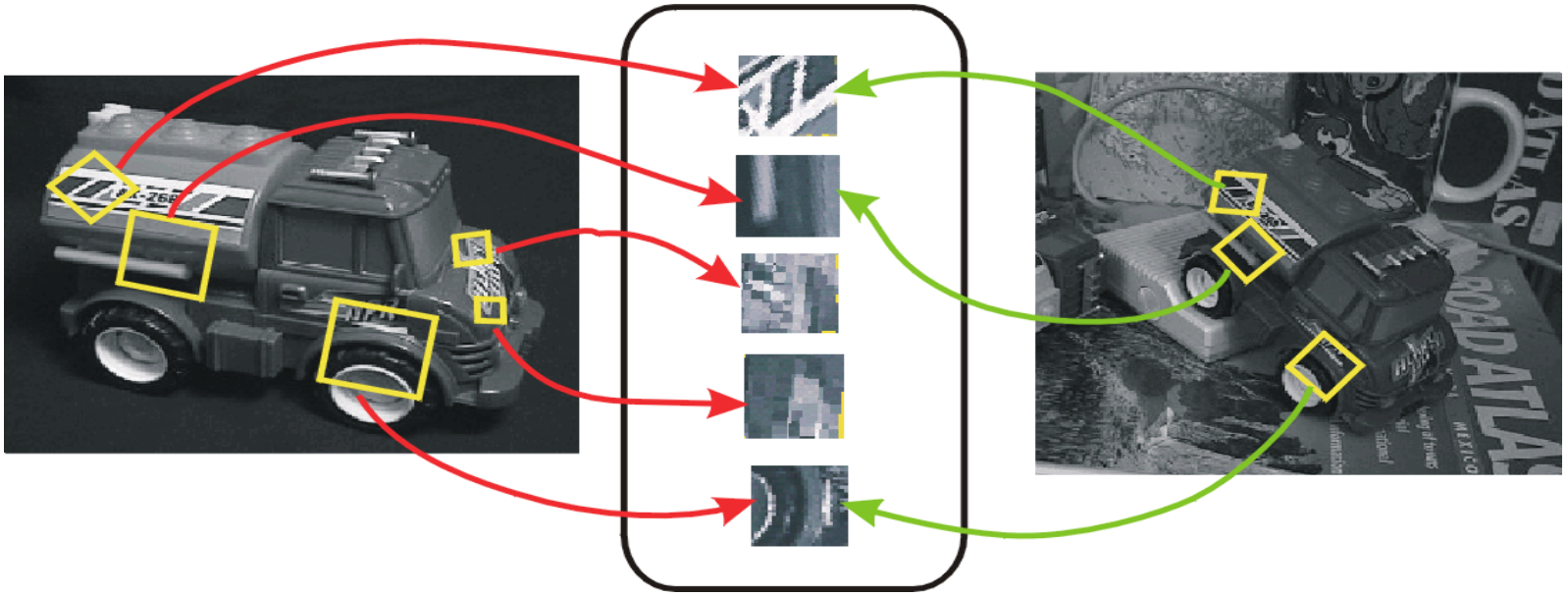
Point descriptor should be:

1. **Invariant**
2. **Distinctive**

Slide credit: Kristen Grauman

JACOBS UNIVERSITY

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



Following slides credit: CVPR 2003 Tutorial on **Recognition and Matching Based on Local Invariant Features** David Lowe

ADVANTAGES OF INVARIANT LOCAL FEATURES

JACOBS UNIVERSITY

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

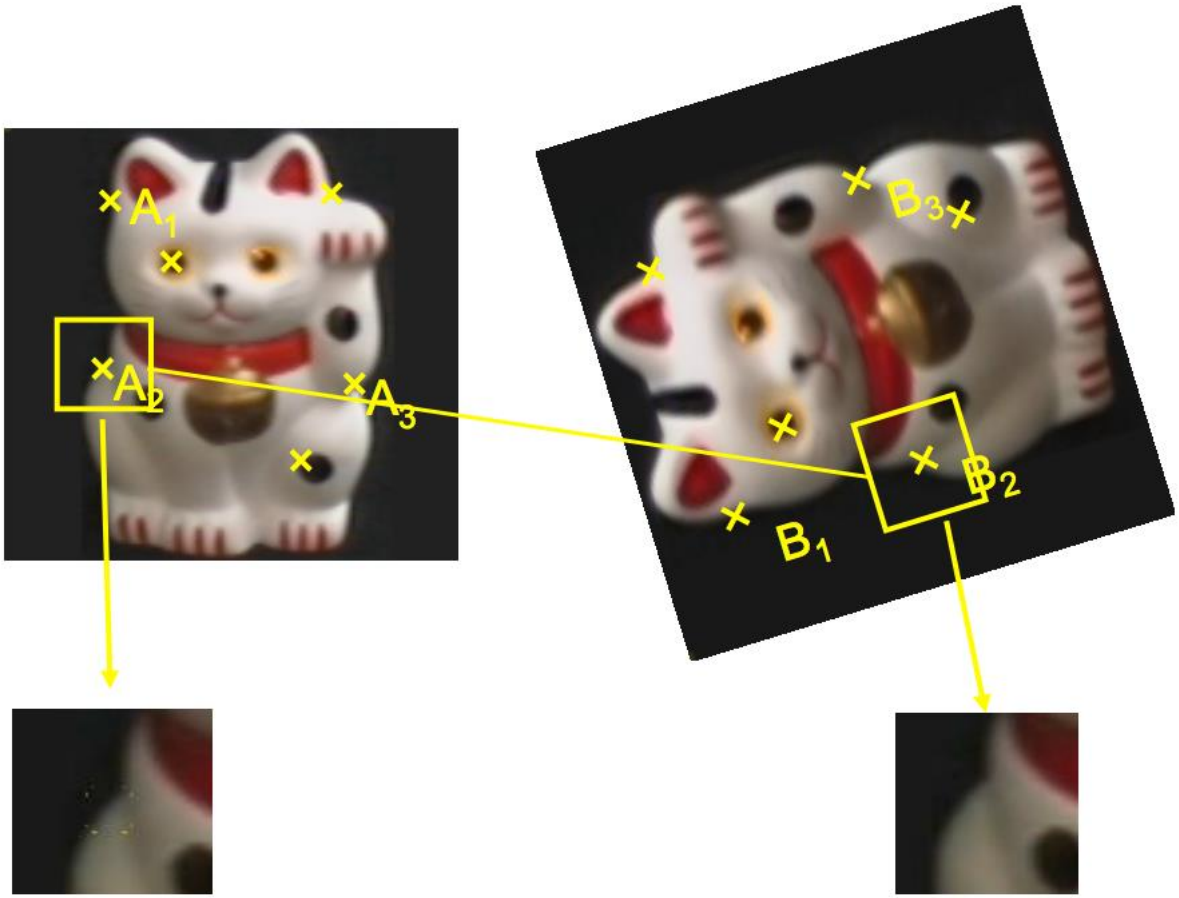
2. **Distinctiveness:** individual features can be matched to a large database of objects

3. **Quantity:** many features can be generated for even small objects

4. **Efficiency:** close to real-time performance

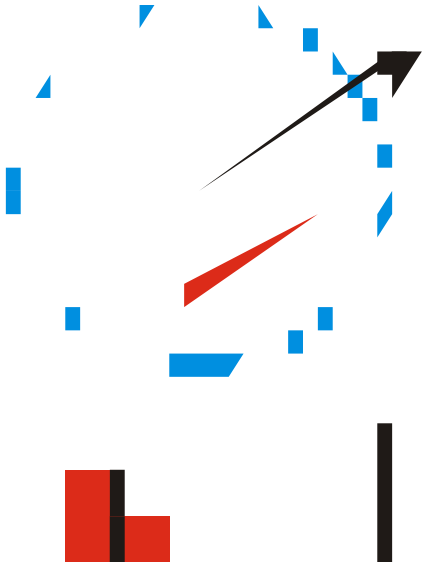
5. **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

ROTATION INVARIANCE?

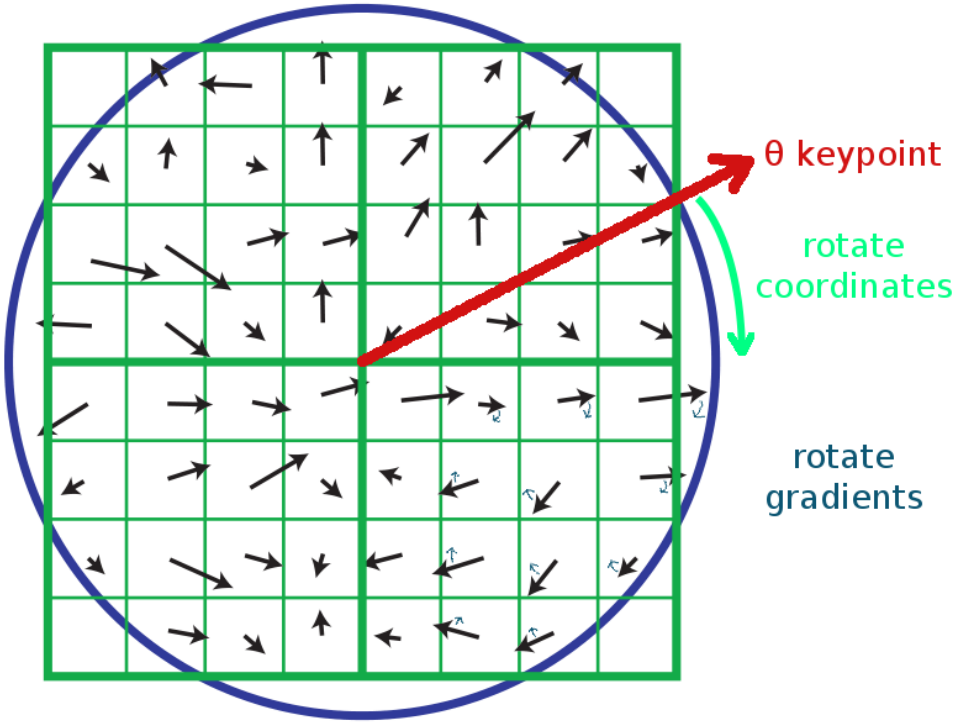


ROTATION INVARIANCE?

1. We are given a keypoint and its scale from DoG
2. We will select a characteristic orientation for the keypoint (based on the most prominent gradient there; discussed next slide)
3. We will describe all features **relative** to this orientation
4. Causes features to be rotation invariant!
   - If the keypoint appears rotated in another image, the features will be the same, because they're **relative** to the characteristic orientation

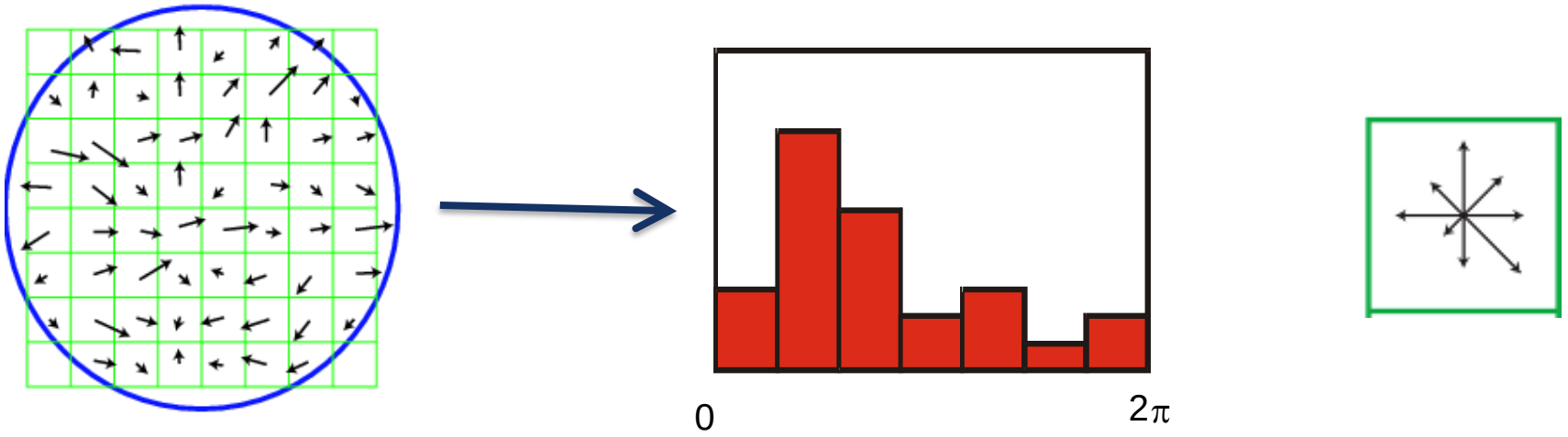


SIFT DESCRIPTOR FORMATION



1. Use the blurred image associated with the keypoint's scale
2. Take image gradients over the keypoint neighborhood.
3. To become rotation invariant, rotate the gradient directions AND locations by (-keypoint orientation)
   - Now we've cancelled out rotation and have gradients expressed at locations **relative** to keypoint orientation θ
   - We could also have just rotated the whole image by -θ, but that would be slower.

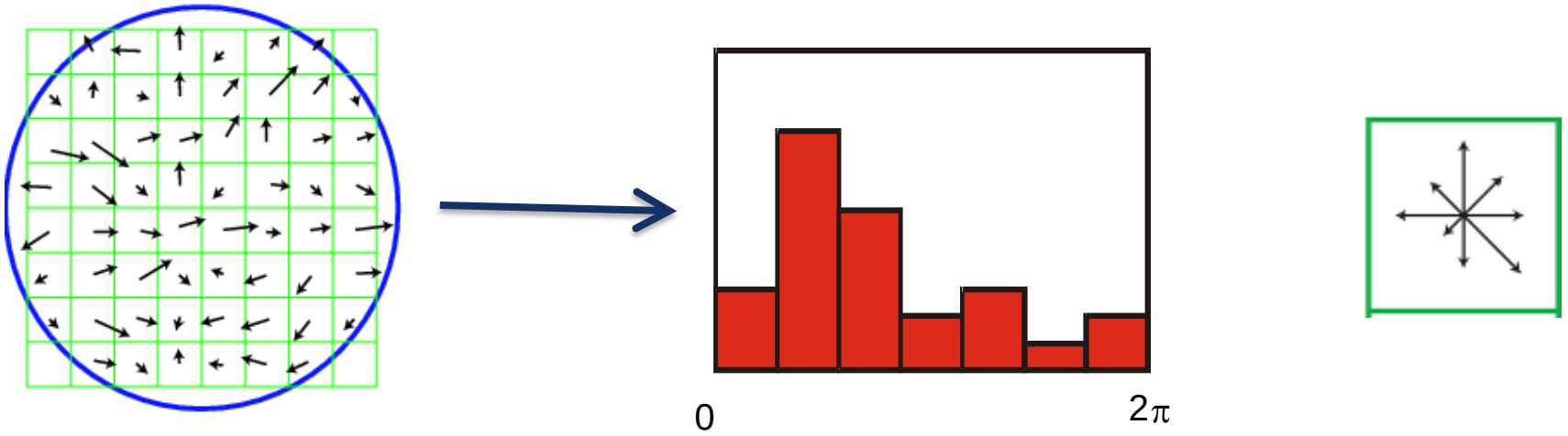
SIFT DESCRIPTOR FORMATION



1. Using precise gradient locations is fragile. We'd like to allow some "slop" in the image, and still produce a very similar descriptor
2. Create array of orientation histograms (a 4x4 array is shown)
3. Put the rotated gradients into their local orientation histograms
   - A gradient's contribution is divided among the nearby histograms based on distance. If it's halfway between two histogram locations, it gives a half contribution to both.
   - Also, scale down gradient contributions for gradients far from the center

The SIFT authors found that best results were with 8 orientation bins per histogram.

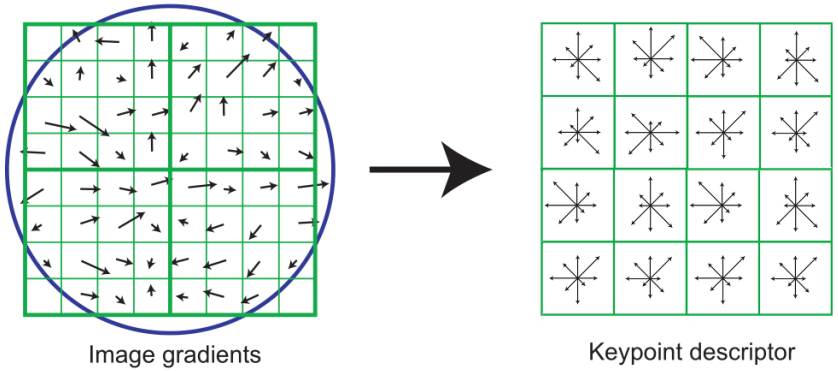
SIFT DESCRIPTOR FORMATION



1. Using precise gradient locations is fragile. We'd like to allow some "slop" in the image, and still produce a very similar descriptor
2. Create array of orientation histograms (a 4x4 array is shown)
3. Put the rotated gradients into their local orientation histograms
   - A gradient's contribution is divided among the nearby histograms based on distance. If it's halfway between two histogram locations, it gives a half contribution to both.
   - Also, scale down gradient contributions for gradients far from the center

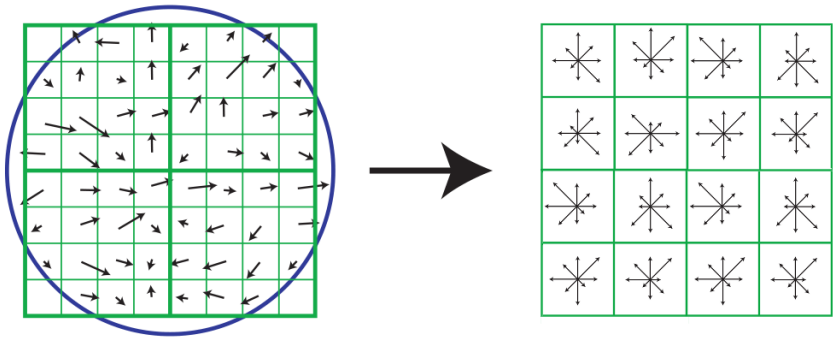
The SIFT authors found that best results were with 8 orientation bins per histogram.

SIFT DESCRIPTOR FORMATION



1. 8 orientation bins per histogram, and a 4x4 histogram array, yields 8 x 4x4 = 128 numbers.
2. So a SIFT descriptor is a length 128 vector, which is invariant to rotation (because we rotated the descriptor) and scale (because we worked with the scaled image from DoG)
3. We can compare each vector from image A to each vector from image B to find matching keypoints!
   - Euclidean "distance" between descriptor vectors gives a good measure of keypoint similarity

SIFT DESCRIPTOR FORMATION



1. Adding robustness to illumination changes:
2. Remember that the descriptor is made of gradients (differences between pixels), so it's already invariant to changes in brightness (e.g. adding 10 to all image pixels yields the exact same descriptor)
3. A higher-contrast photo will increase the magnitude of gradients linearly. So, to correct for contrast changes, normalize the vector (scale to length 1.0)
4. Very large image gradients are usually from unreliable 3D illumination effects (glare, etc). So, to reduce their effect, clamp all values in the vector to be ≤ 0.2 (an experimentally tuned value). Then normalize the vector again.
5. Result is a vector which is fairly invariant to illumination changes.

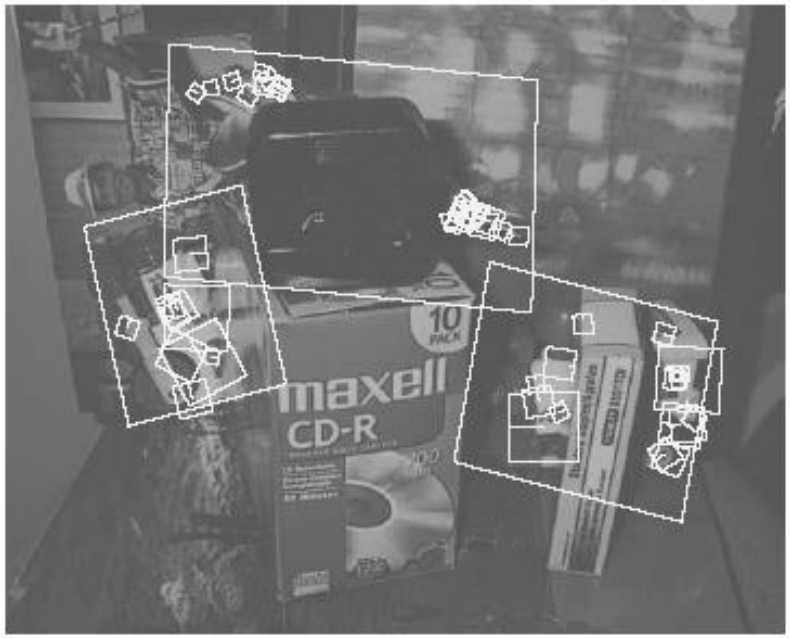


Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affi ne transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

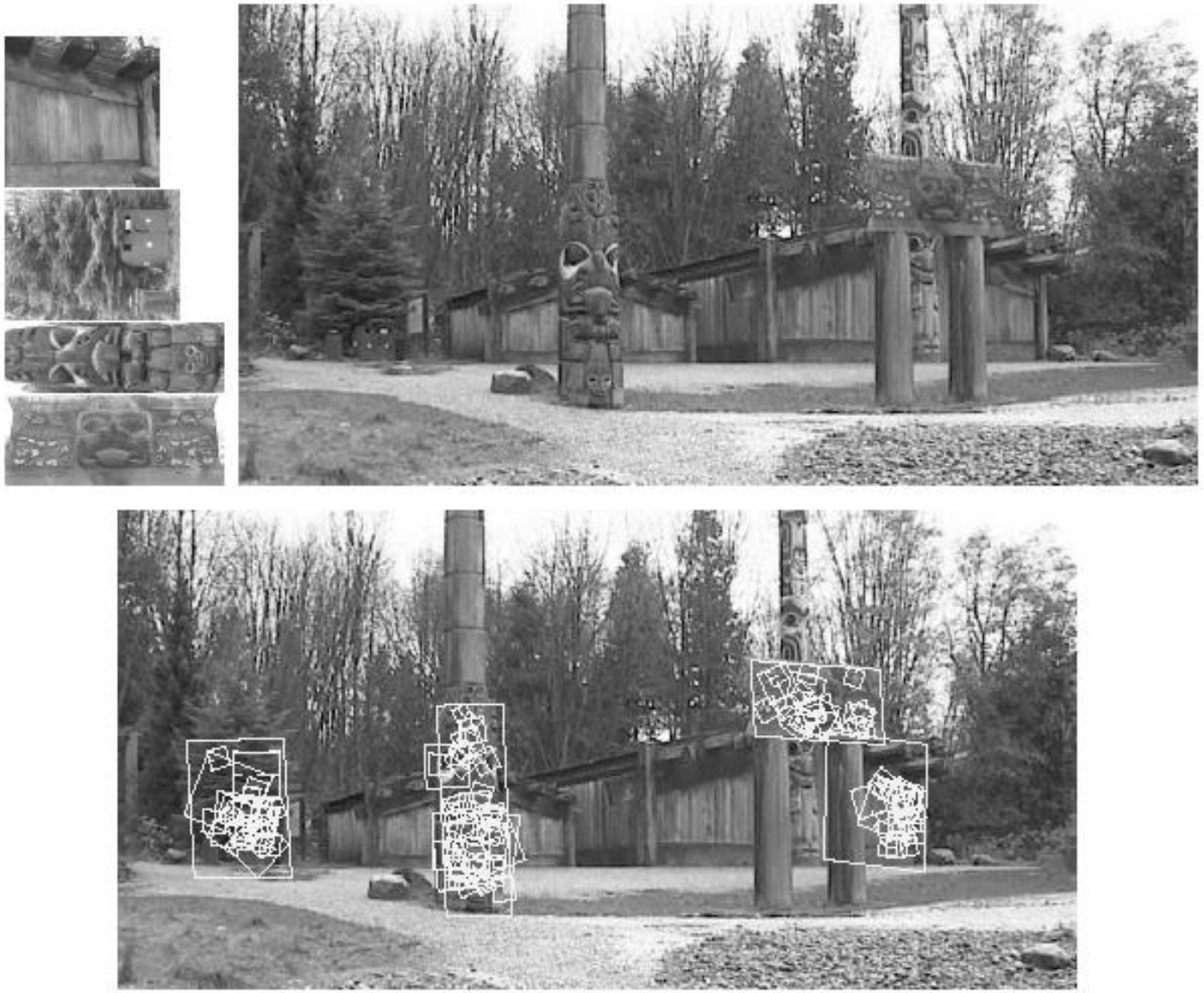


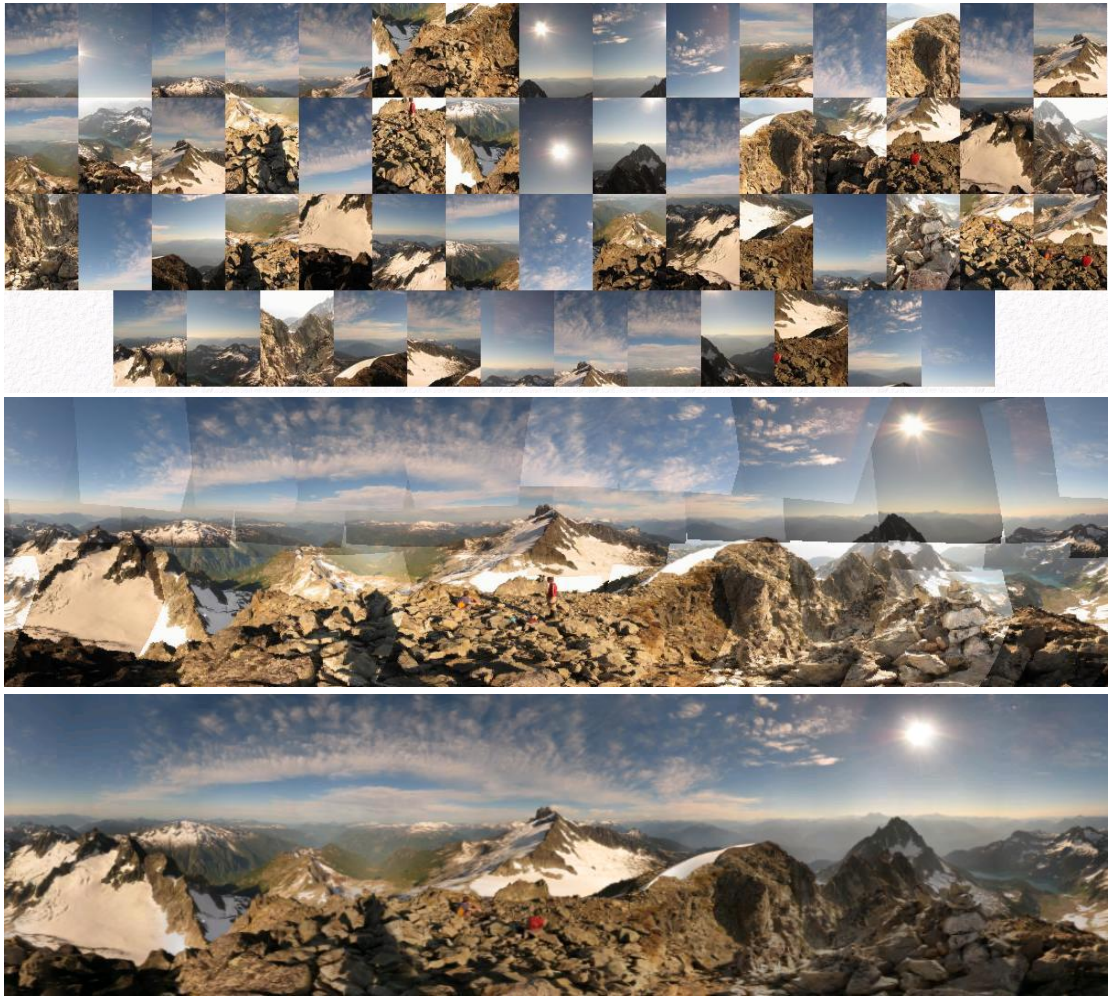
Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affi ne transform used for recognition.

NICE SIFT RESOURCES

- an online tutorial: http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/

- Wikipedia: http://en.wikipedia.org/wiki/Scale-invariant_feature_transform

AUTOMATIC MOSAICING



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

[Image from T. Tuytelaars ECCV 2006 tutorial]

APPLICATIONS OF LOCAL INVARIANT FEATURES

JACOBS UNIVERSITY

- Wide baseline stereo
- Motion tracking
- **Panoramas**
- Mobile robot navigation
- 3D reconstruction
- Recognition
- …

WHAT WE WILL LEARN TODAY?

- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HOG: another image descriptor
- Application: Panorama

HISTOGRAM OF ORIENTED GRADIENTS
(NOT IN THE FINAL EXAM)

JACOBS UNIVERSITY

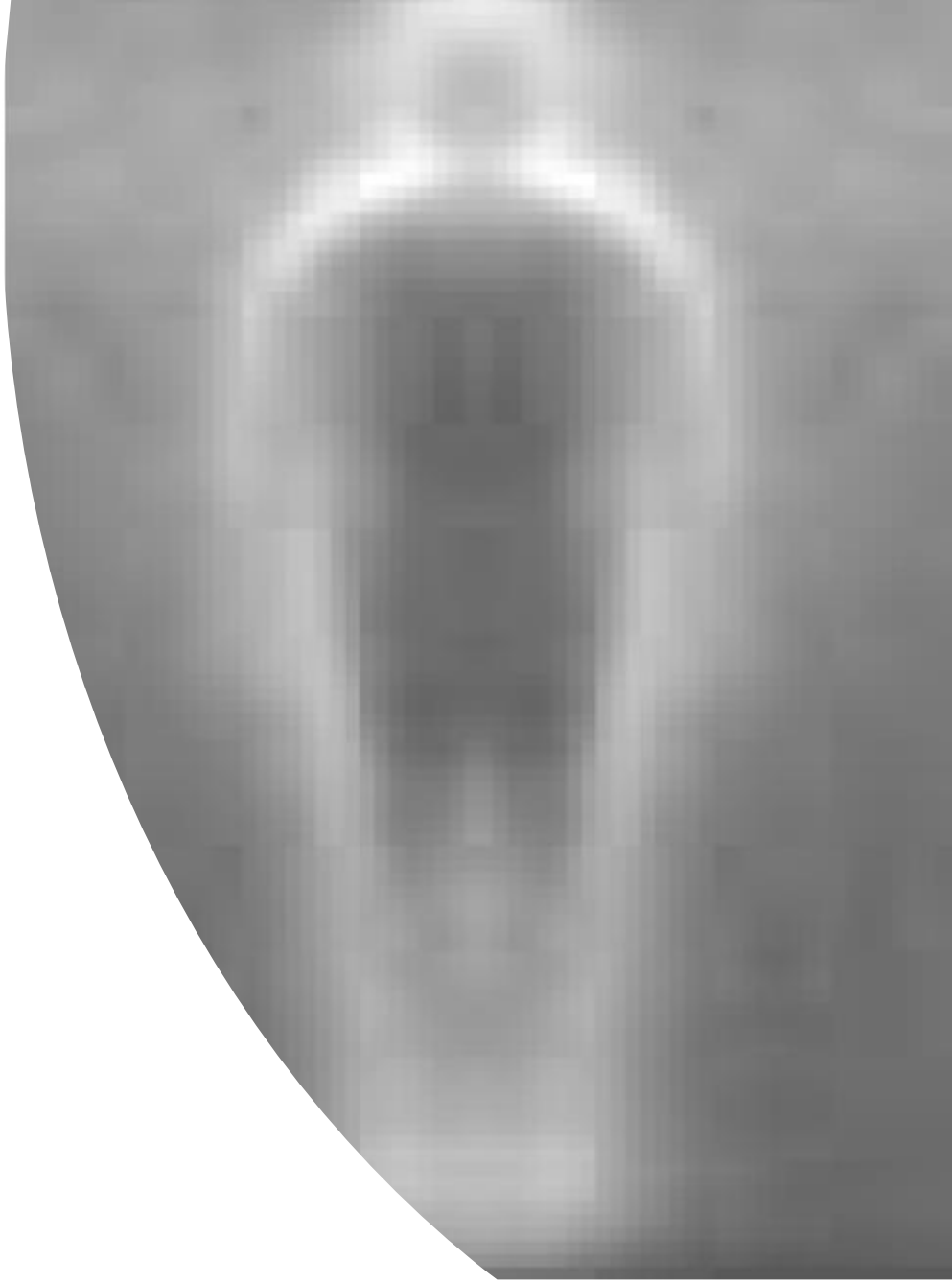
- Find robust feature set that allows object form to be discriminated.

- Challenges
  - Wide range of pose and large variations in appearances
  - Cluttered backgrounds under different illumination
  - “Speed” for mobile vision

- References
  - [1] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, pages 886-893, 2005
  - [2] Chandrasekhar et al. CHoG: Compressed Histogram of Gradients - A low bit rate feature descriptor, CVPR 2009

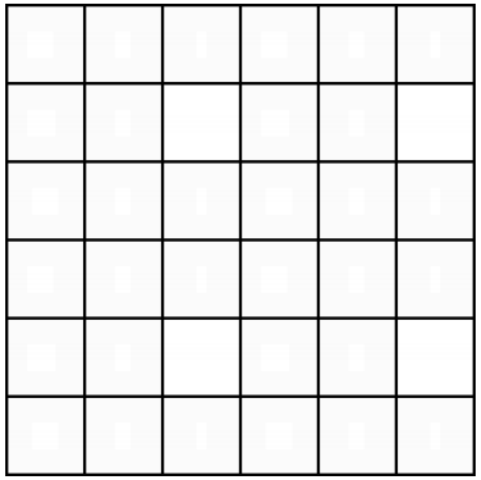
HISTOGRAM OF ORIENTED GRADIENTS

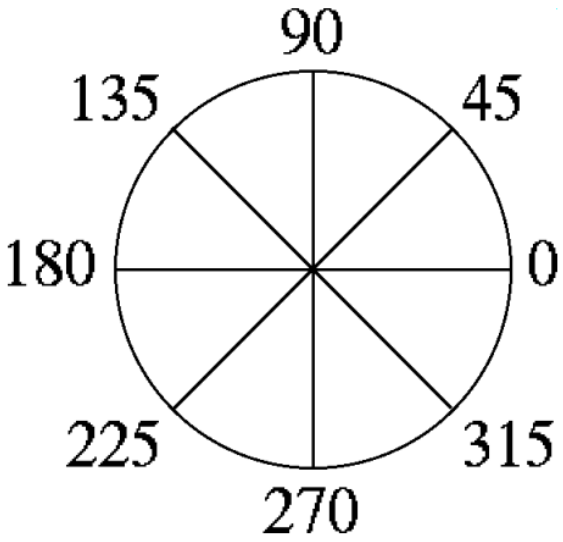
Local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions.



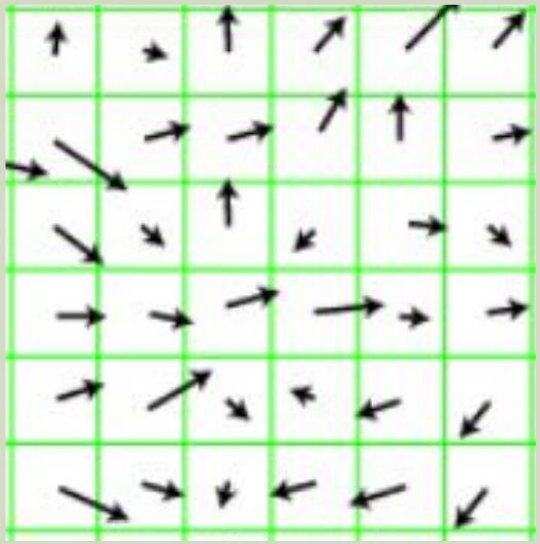
HISTOGRAM OF ORIENTED GRADIENTS

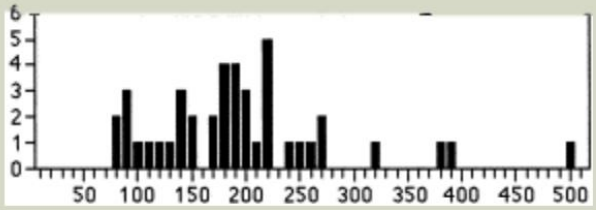
1. Dividing the image window into small spatial regions (cells)
2. Cells can be either rectangle or radial.
3. Each cell accumulating a weighted local 1-D histogram of gradient directions over the pixels of the cell.



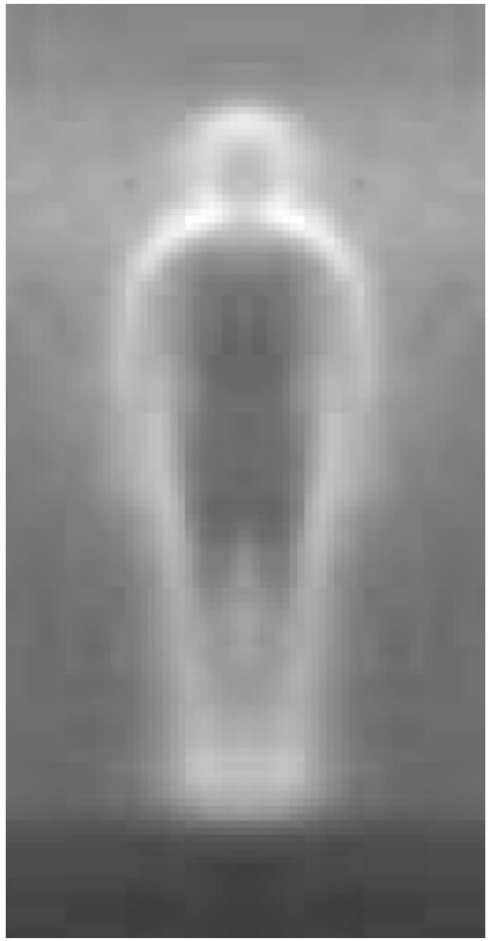


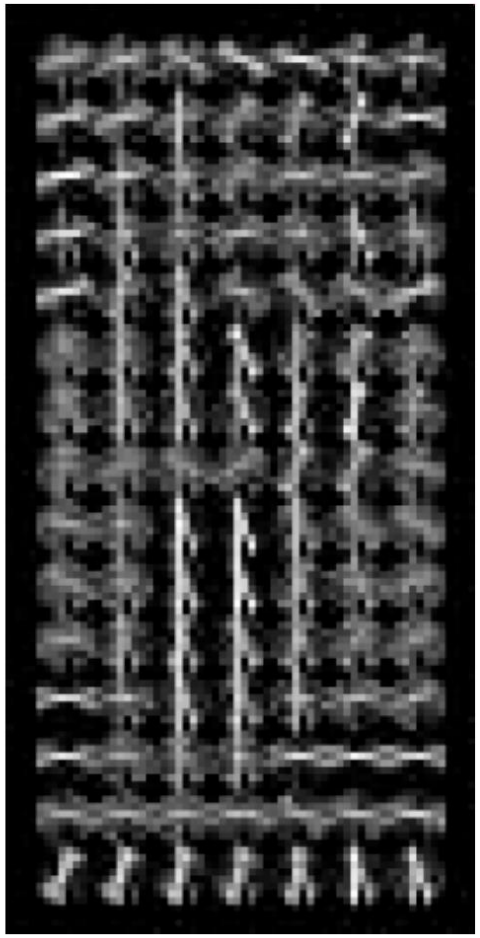
HISTOGRAM OF ORIENTED GRADIENTS





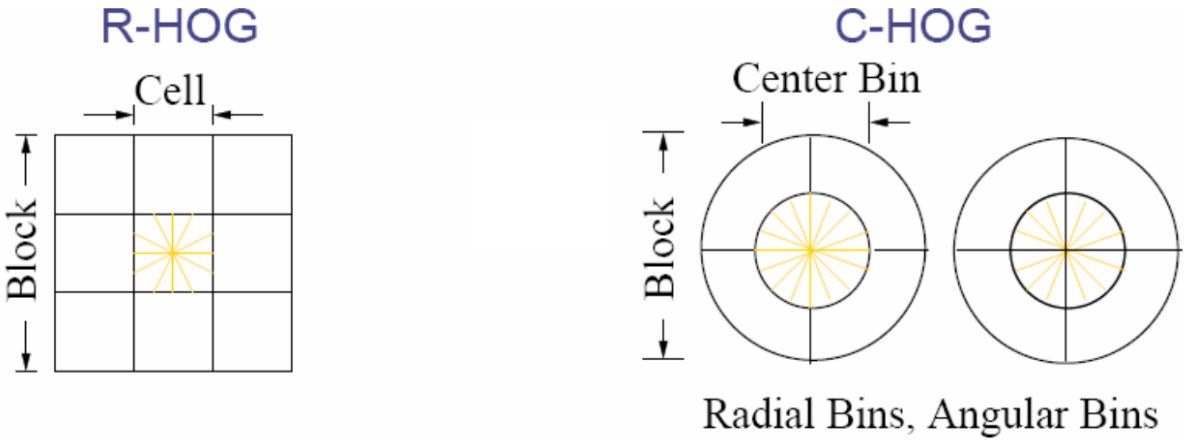
HISTOGRAM OF ORIENTED GRADIENTS



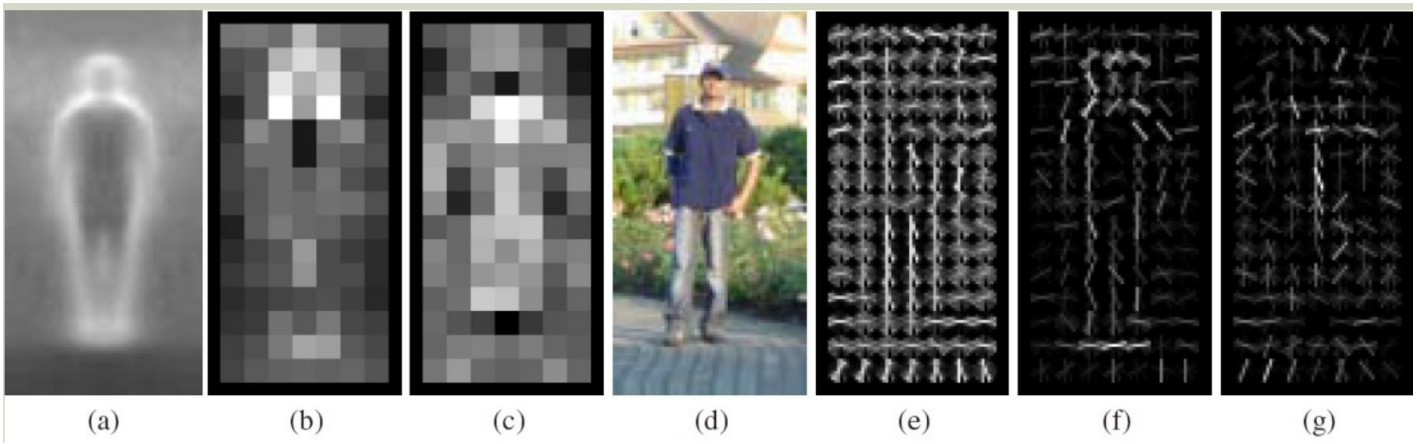


NORMALIZATION

1. For better invariance to illumination and shadowing. it is useful to contrast-normalize the local responses before using them.
2. Accumulate local histogram “energy” over a larger regions (“blocks”) to normalize all of the cells in the block.

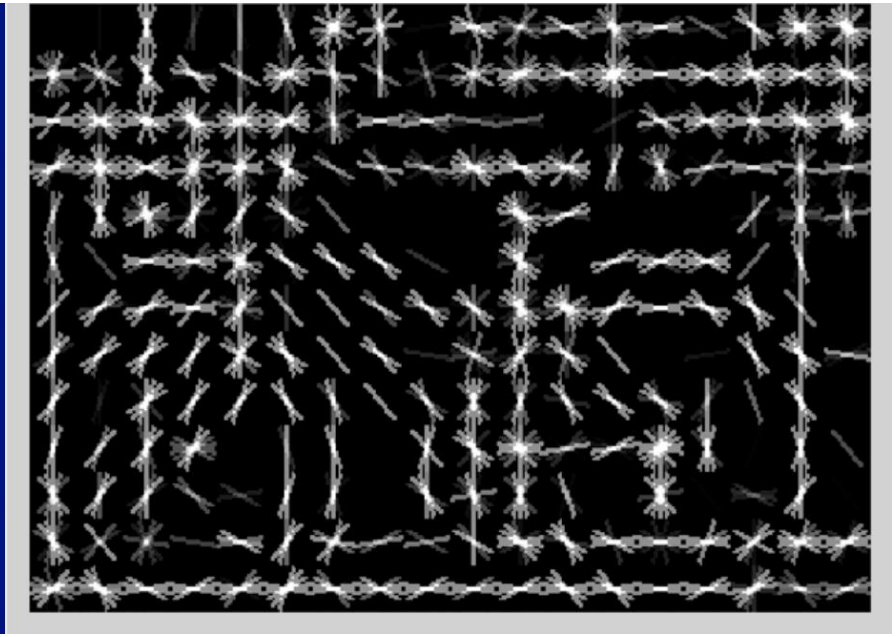


VISUALIZING HOG



a. Average gradient over positive examples
b. Maximum positive weight in each block
c. Maximum negative weight in each block
d. A test image
e. Its R-HOG descriptor
f. R-HOG descriptor weighted by positive weights
g. R-HOG descriptor weighted by negative weights

VISUALIZING HOG



DIFFERENCE BETWEEN HOG AND SIFT

JACOBS UNIVERSITY

1. HoG is usually used to describe entire images. SIFT is used for key point matching
2. SIFT histrograms are oriented towards the dominant gradient. HoG is not.
3. HoG gradients are normalized using neighborhood bins.
4. SIFT descriptors use varying scales to compute multiple descriptors.

WHAT WE WILL LEARN TODAY?

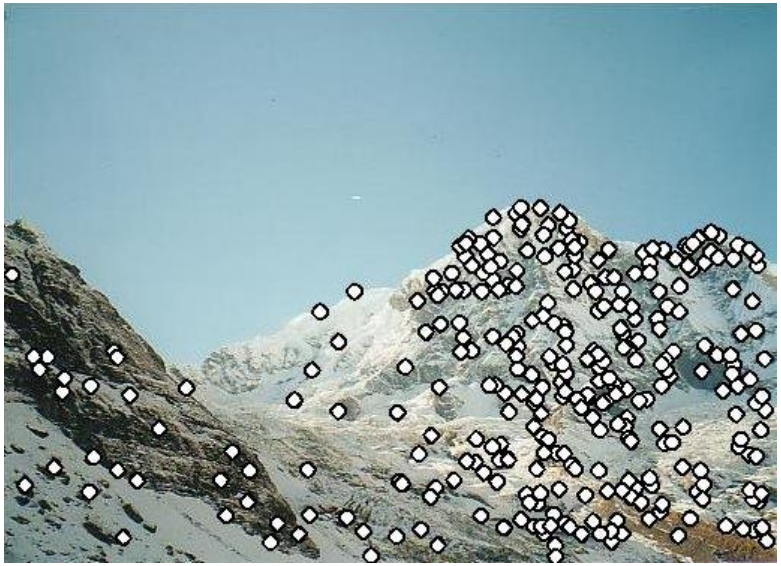
JACOBS UNIVERSITY

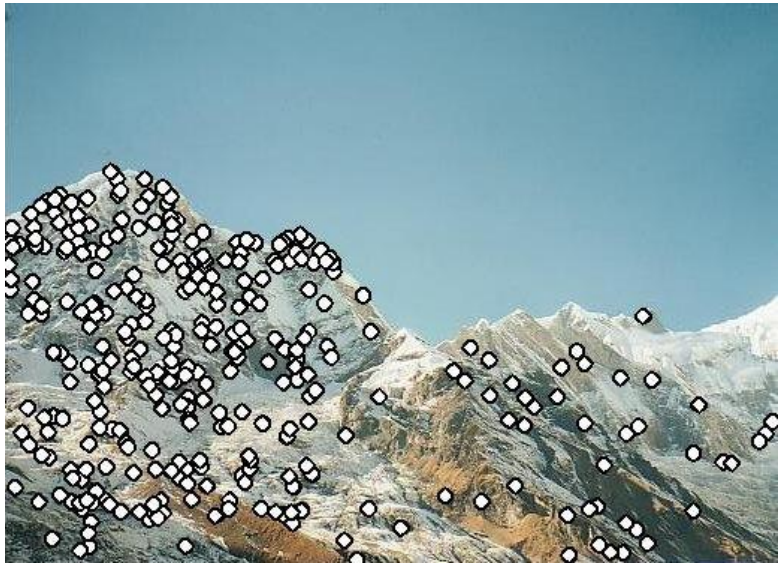
- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HOG: another image descriptor
- Application: Panorama

APPLICATION: IMAGE STITCHING

Slide credit: Darya Frolova, Denis Simakov

APPLICATION: IMAGE STITCHING

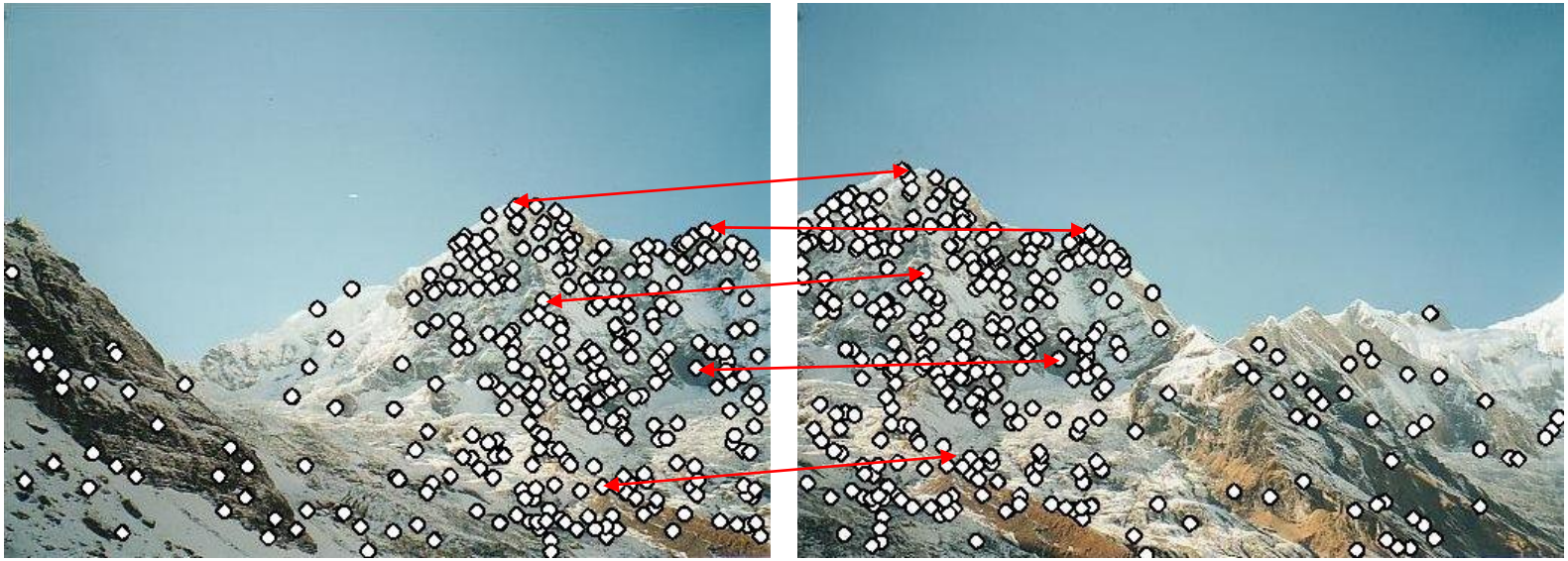




1. Procedure:
   - Detect feature points in both images

Slide credit: Darya Frolova, Denis Simakov

APPLICATION: IMAGE STITCHING



1. Procedure:

- Detect feature points in both images
- Find corresponding pairs

Slide credit: Darya Frolova, Denis Simakov

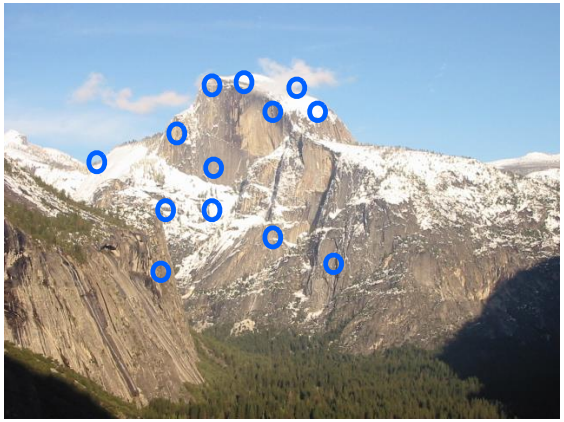
APPLICATION: IMAGE STITCHING

1. Procedure:

- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align the images

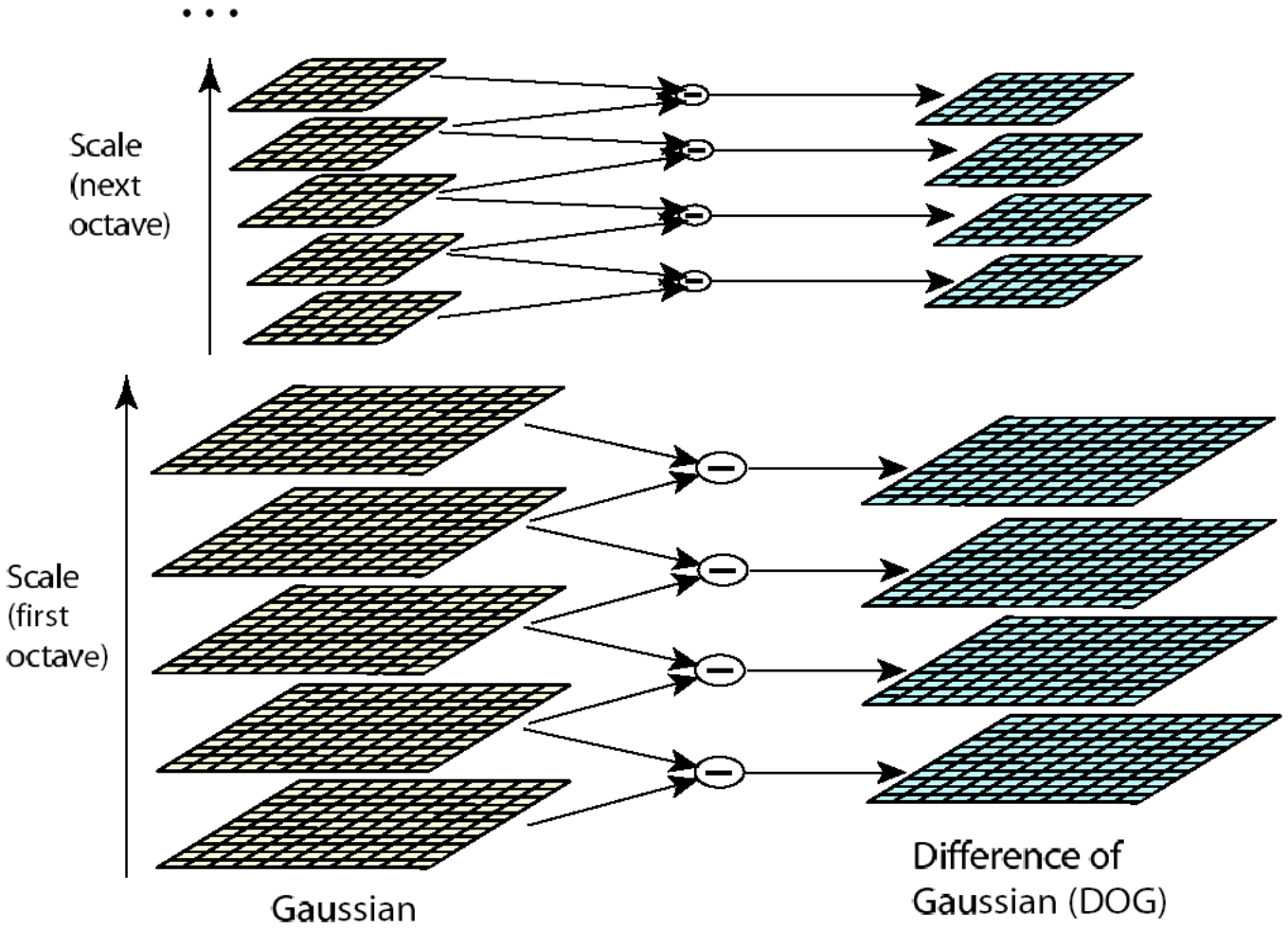
Slide credit: Darya Frolova, Denis Simakov

MAIN FLOW

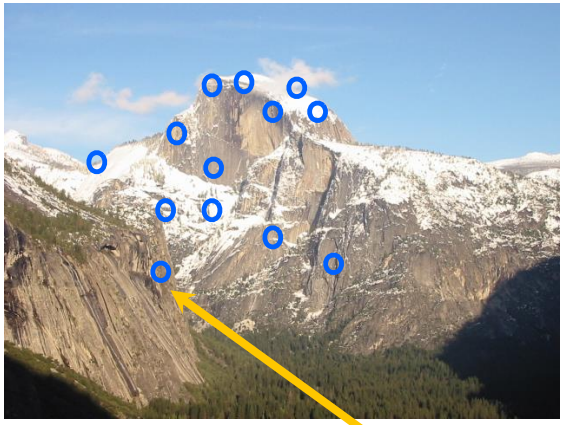


- Detect key points

DETECT KEY POINTS



MAIN FLOW



$(u_1, u_2, \ldots, u_{128})$

$(v_1, v_2, \ldots, v_{128})$

- Detect key points
- Build the SIFT descriptors

BUILD THE SIFT DESCRIPTORS

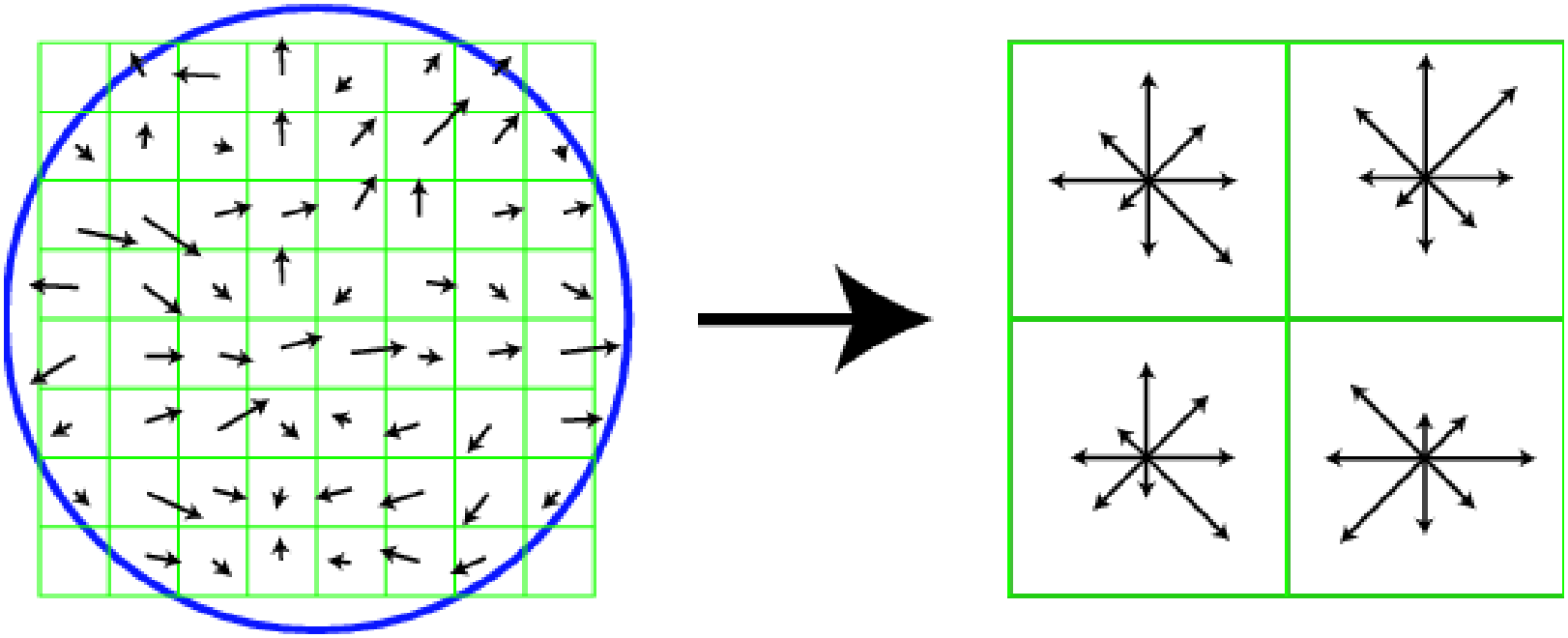
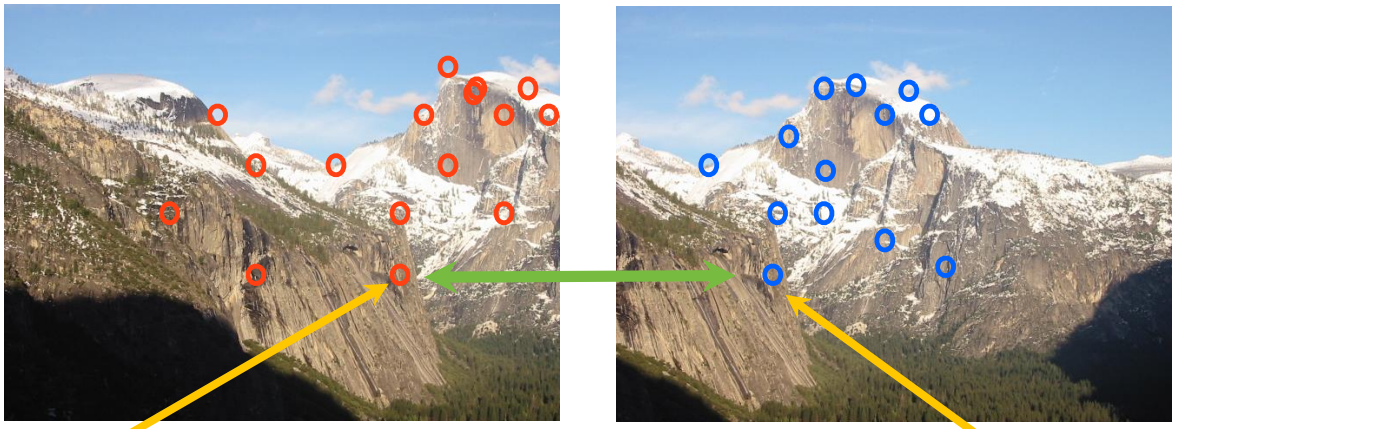


Image gradients

Keypoint descriptor

MAIN FLOW



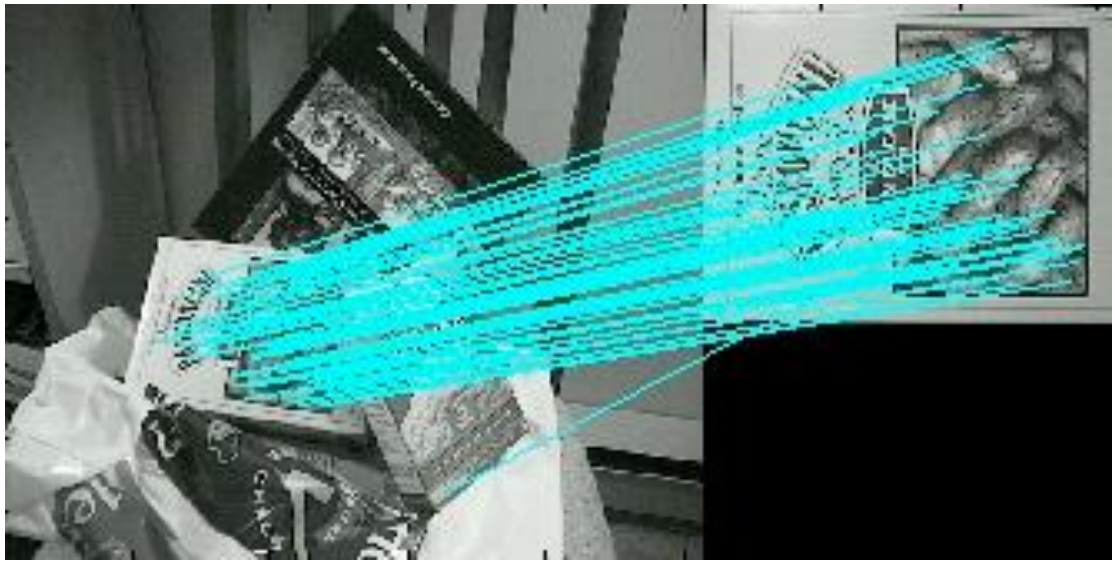
$(u_1, u_2, \ldots, u_{128})$

$(v_1, v_2, \ldots, v_{128})$

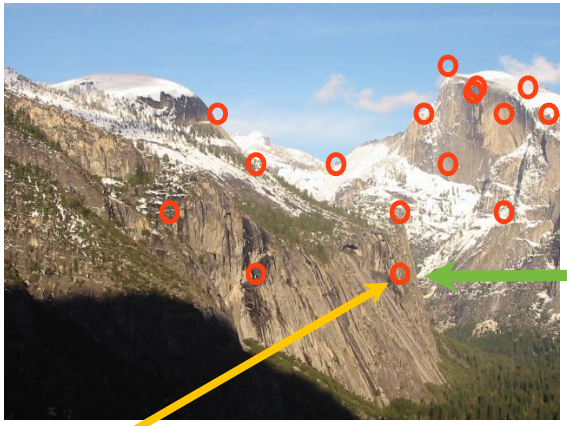
- Detect key points
- Build the SIFT descriptors
- Match SIFT descriptors

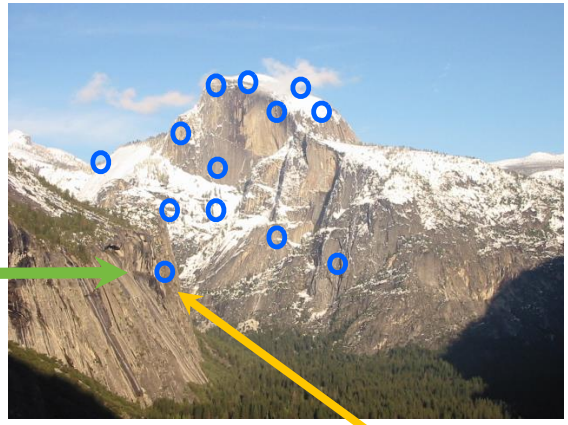
JACOBS UNIVERSITY

Euclidean distance between descriptors



MAIN FLOW





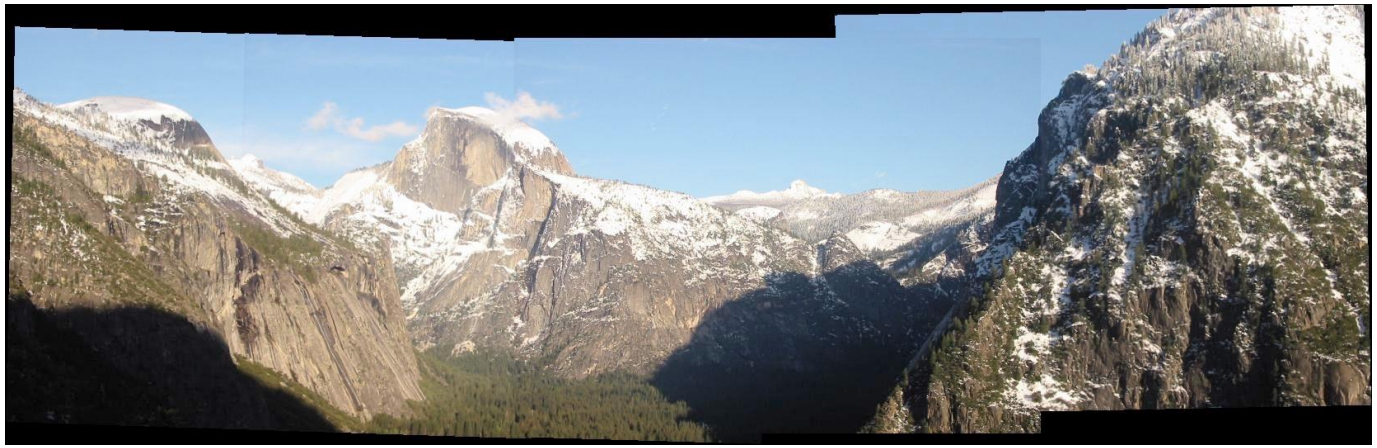
$(u_1, u_2, \ldots, u_{128})$

$(v_1, v_2, \ldots, v_{128})$

- Detect key points
- Build the SIFT descriptors
- Match SIFT descriptors
- Fitting the transformation

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

RESULTS



RESULTS