CS484 - CS555: Introduction to Computer Vision

(Hand-crafted) Features: SIFT

Dr. Sedat Ozer

SIFT:
Scale-Invariant Feature Transform

David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

Question: Which one of these two objects is in the image below?

Say yes if you see Train
Say no if you see Frog

Object_A: Train

Object_B: Frog

Image source: [Lowe 2004]

Occlusion and affine transformation is handled in SIFT

Object_A: Train

Object_B: Frog

Image source: [Lowe 2004]

Question 2: Find the locations of the given 4 images in the larger image

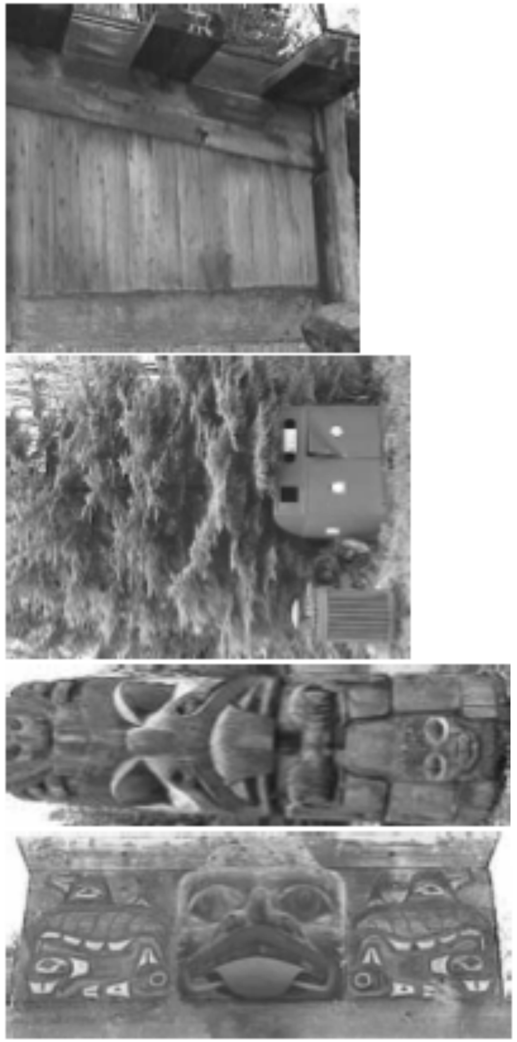
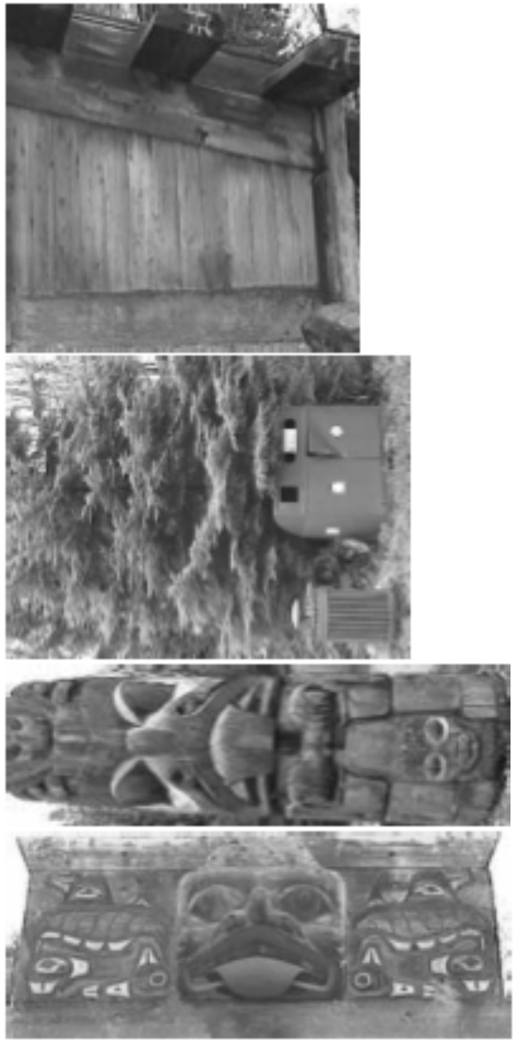


Image source: [Lowe 2004]

Solution: Sift is rotation and scale invariant…



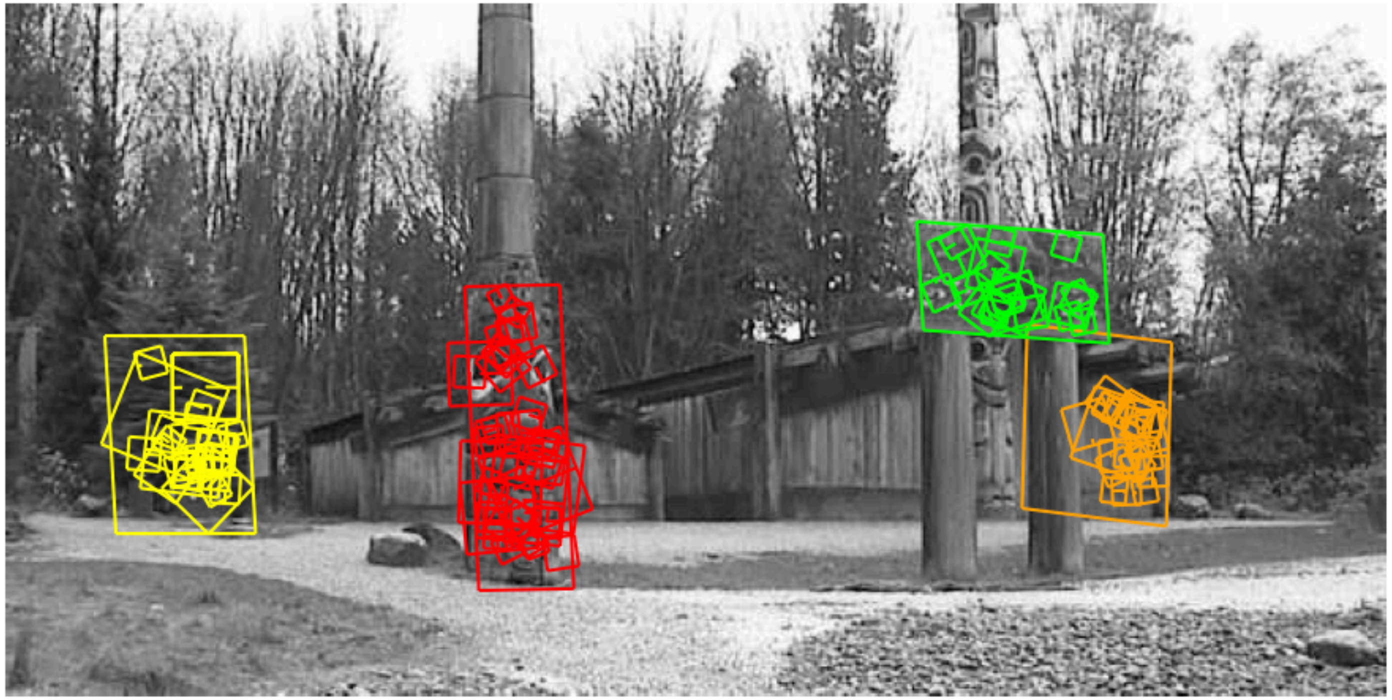


Image source: [Lowe 2004]

Image Stitching

Image 1

Image 2

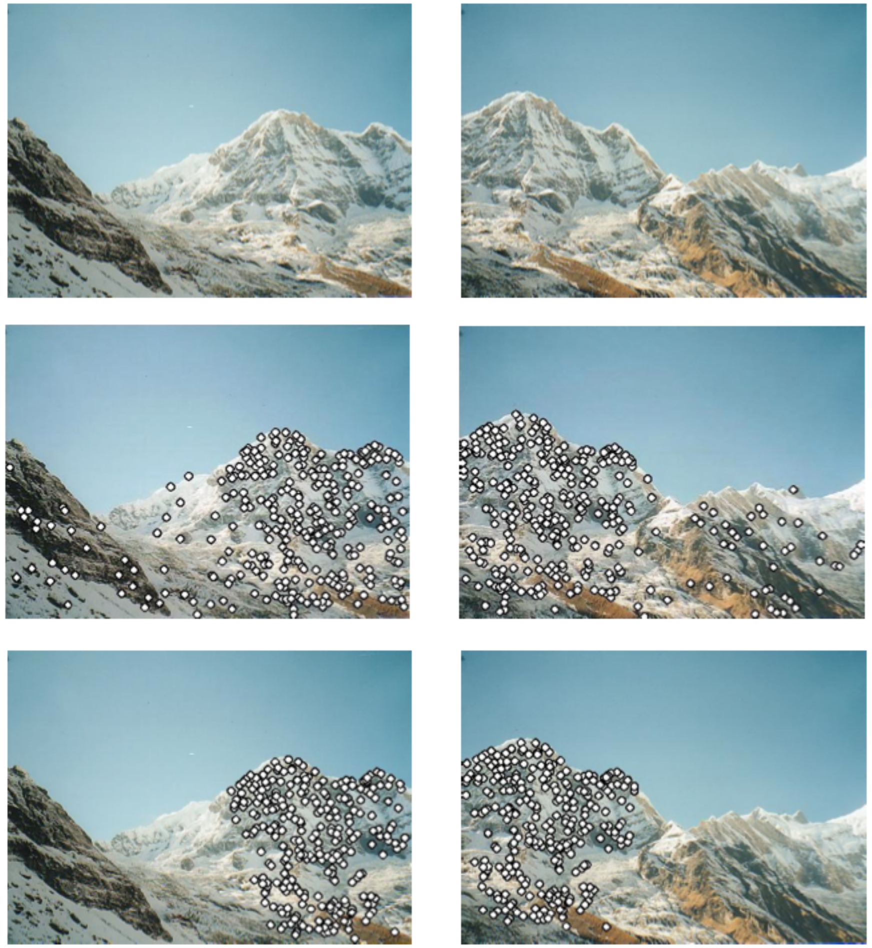
…

How to make one continuous image from these two (or more) separate images?

Image source: M. Brown, and D. G. Lowe. "Automatic panoramic image stitching using invariant features." *International Journal of Computer Vision* 74, no. 1 (2007): 59-73.

Image Stitching

Image 1 Image 2

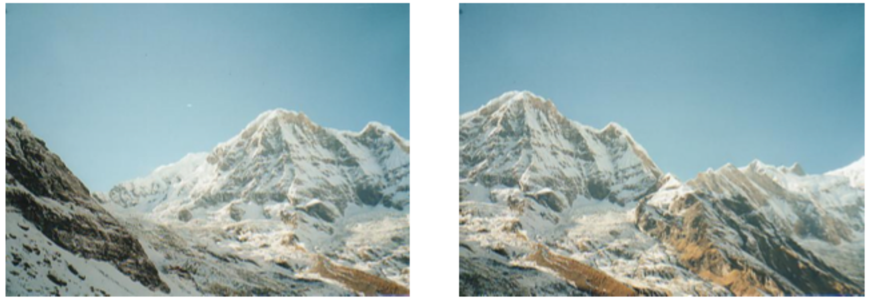


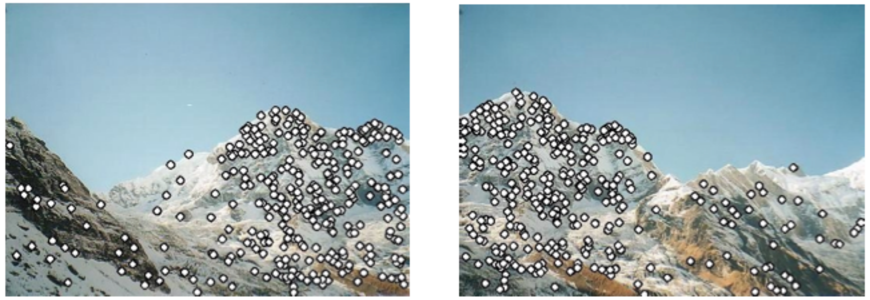
Compute some interesting (key) points

Image source: M. Brown, and D. G. Lowe. "Automatic panoramic image stitching using invariant features." *International Journal of Computer Vision* 74, no. 1 (2007): 59-73.

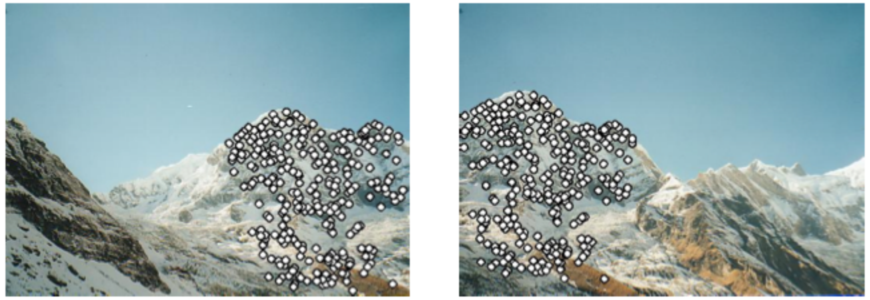
Image Stitching

Image 1 Image 2





Compute some interesting (key) points

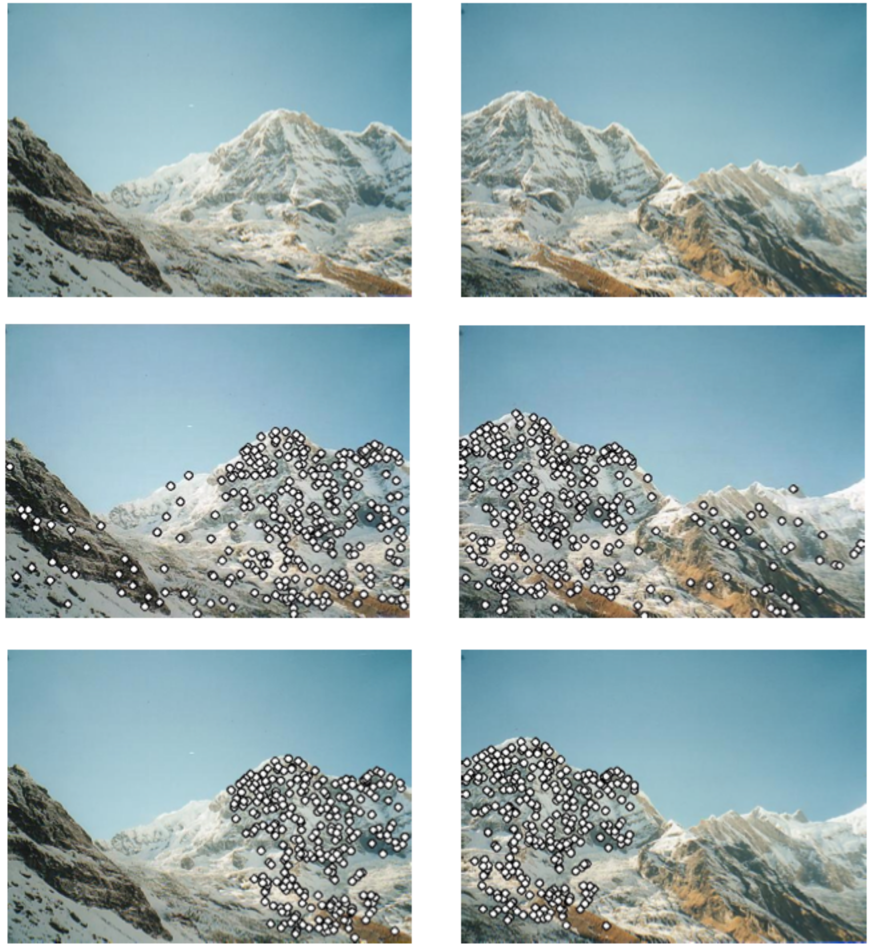


Find the matching key points

Image source: M. Brown, and D. G. Lowe. "Automatic panoramic image stitching using invariant features." *International Journal of Computer Vision* 74, no. 1 (2007): 59-73.

Image Stitching

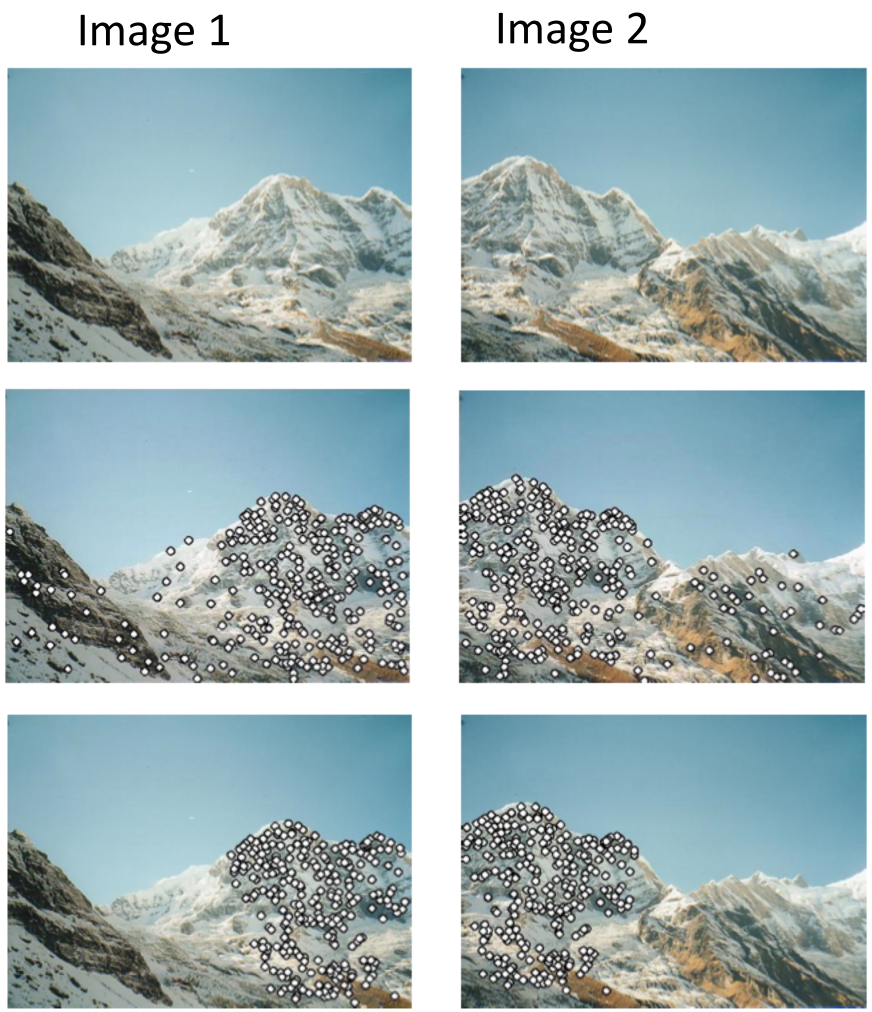
Image 1 Image 2



Then overlap the matching points!

Image source: M. Brown, and D. G. Lowe. "Automatic panoramic image stitching using invariant features." *International Journal of Computer Vision* 74, no. 1 (2007): 59-73.

Image Stitching with SIFT



One of many available techniques is: SIFT

Compute some interesting (key) points

Find the matching key points

Image source: M. Brown, and D. G. Lowe. "Automatic panoramic image stitching using invariant features." *International Journal of Computer Vision* 74, no. 1 (2007): 59-73.

SIFT: Scale-Invariant Feature Transform

- **Find the descriptive local points (find keypoints):**
  1) **scale-space extrema detection :** Find all the extrema points as candidate interest points in scale space,
  2) **Keypoint localization:** For each interest points in scale space, compute the location and the scale.

- **Create a feature vector for each local key-point:**
  3) **Orientation Assignment:** For each keypoint location, compute orientation(s) based on the local image gradients.
  4) **Keypoint descriptor:** Compute local keypoint descriptors using the local image gradients.

SIFT (1): Scale-space extrema detection (1): Scale Space

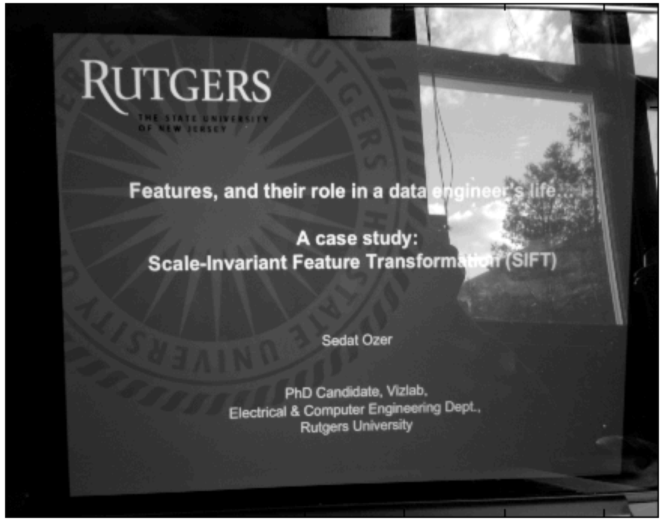
Image source: Iphone ☺

SIFT (1): Scale-space extrema detection (1): Scale Space

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

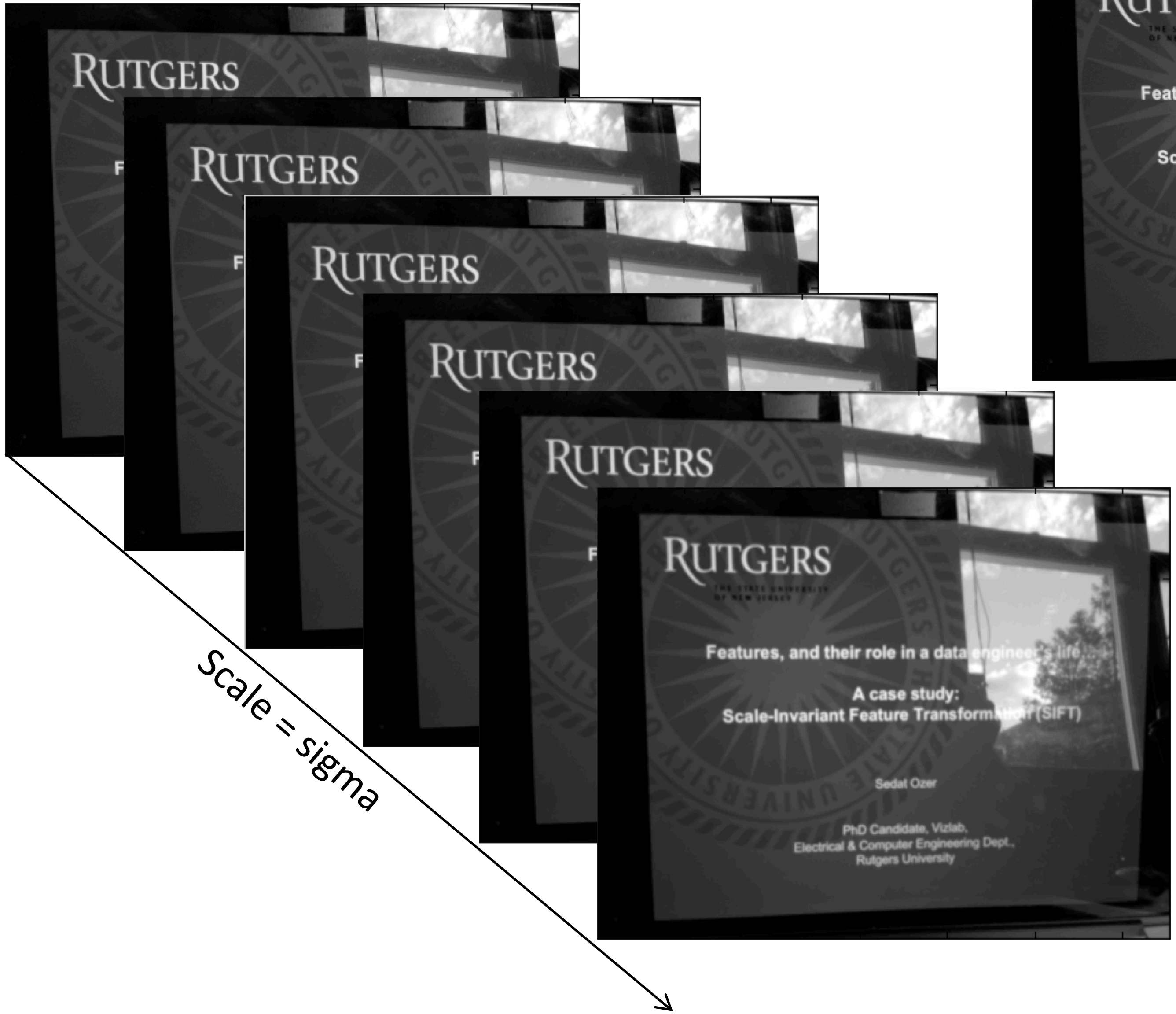
image $I\,(x, y)$ → (Convolution) → Scale Space $L\,(x, y, \sigma)$

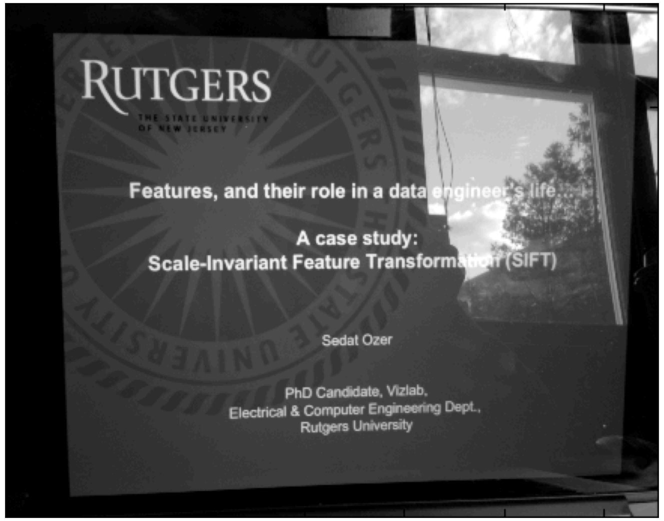
$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$



(Grayscale)

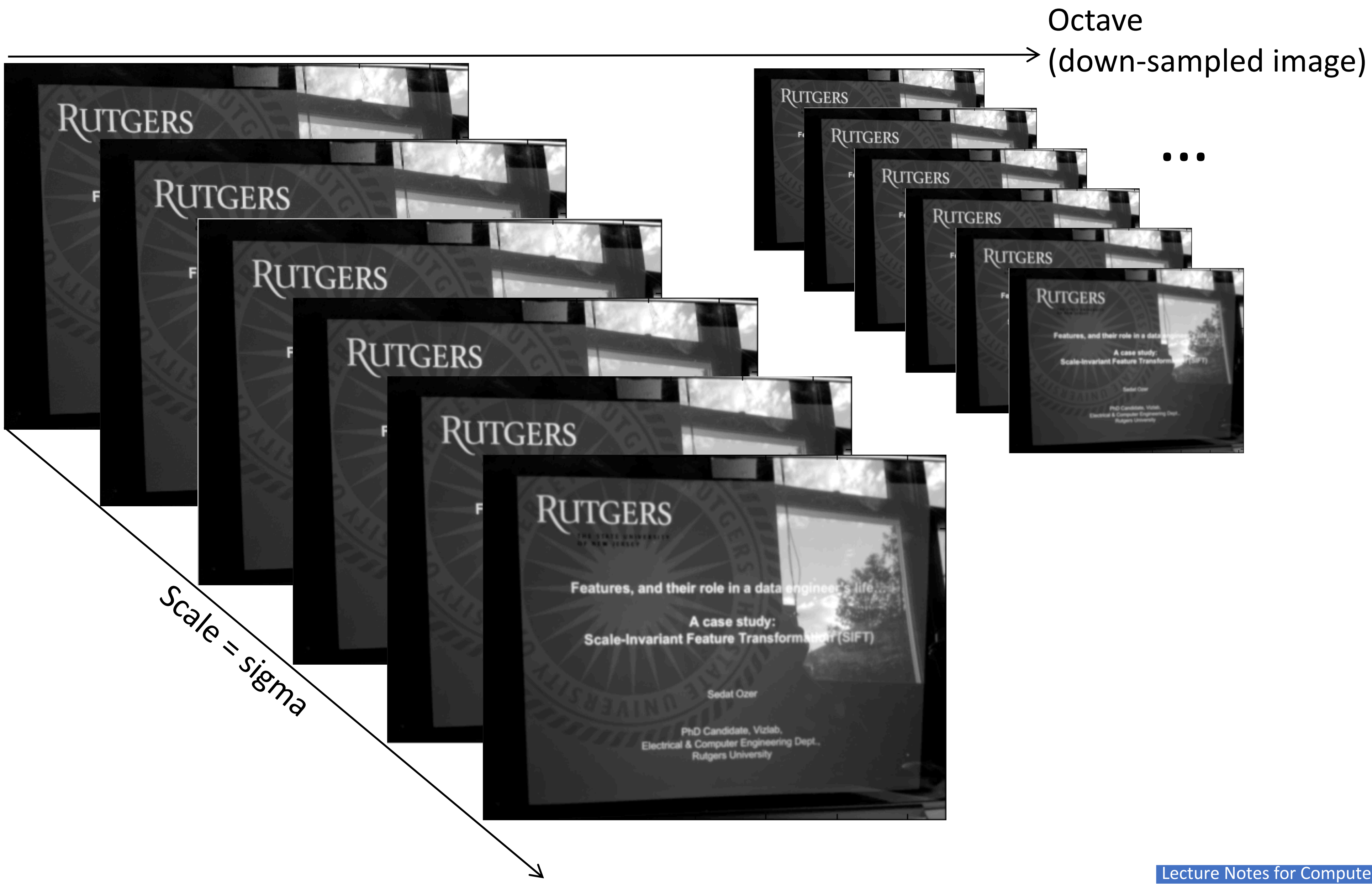
SIFT (1): Scale-space extrema detection (1): Scale Space



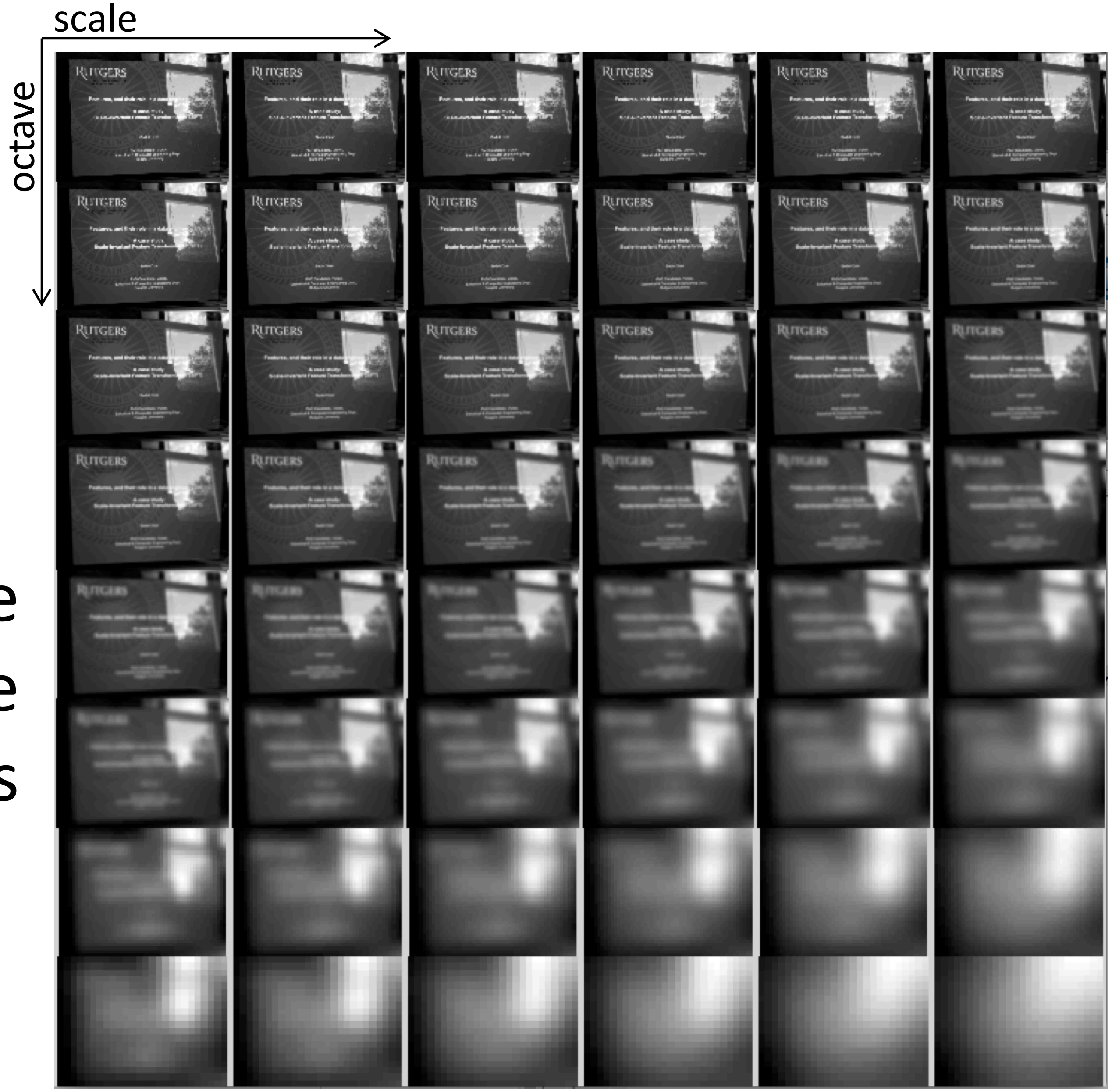


Original image

SIFT (1): Scale-space extrema detection (1): Scale Space



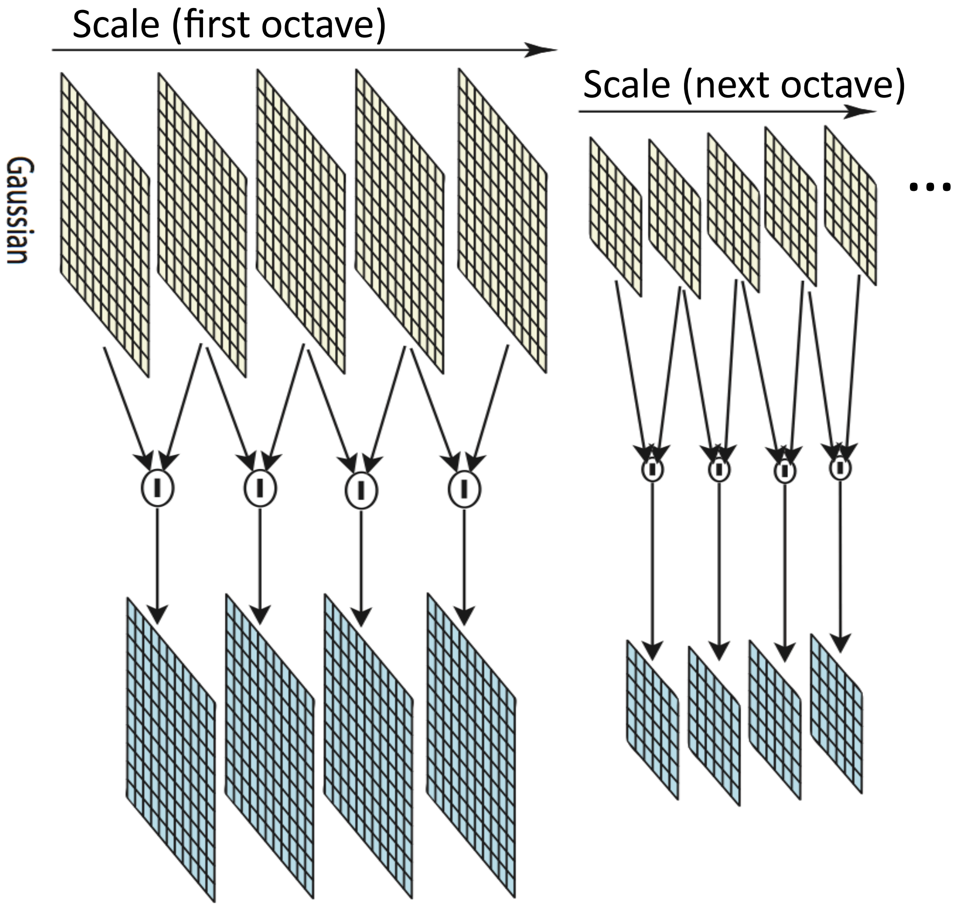
SIFT (1): Gaussian Scale Space



This example has:
8 octave
6 scale
values

SIFT (1) Difference of Gaussian (DoG)

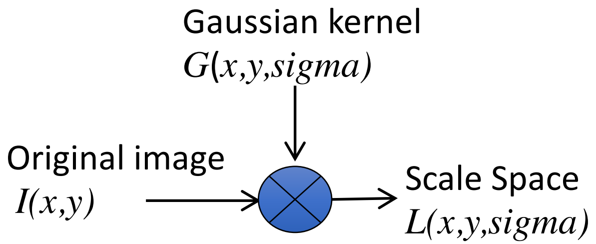
Image source: Lowe (2004)



Difference of Gaussian (DoG):
An approximation for the Laplacian of the Gaussian operator

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

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y),$$



$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

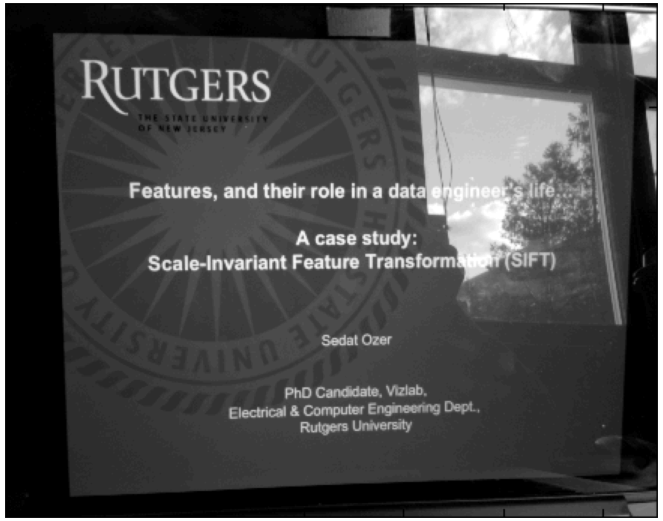
$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2 \nabla^2 G.$$

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma). \end{aligned}$$

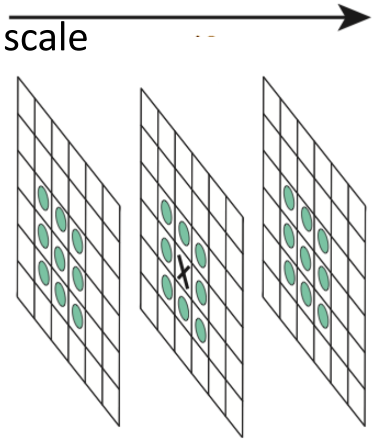
Laplacian of a Gaussian ~ DoG

Approximated!

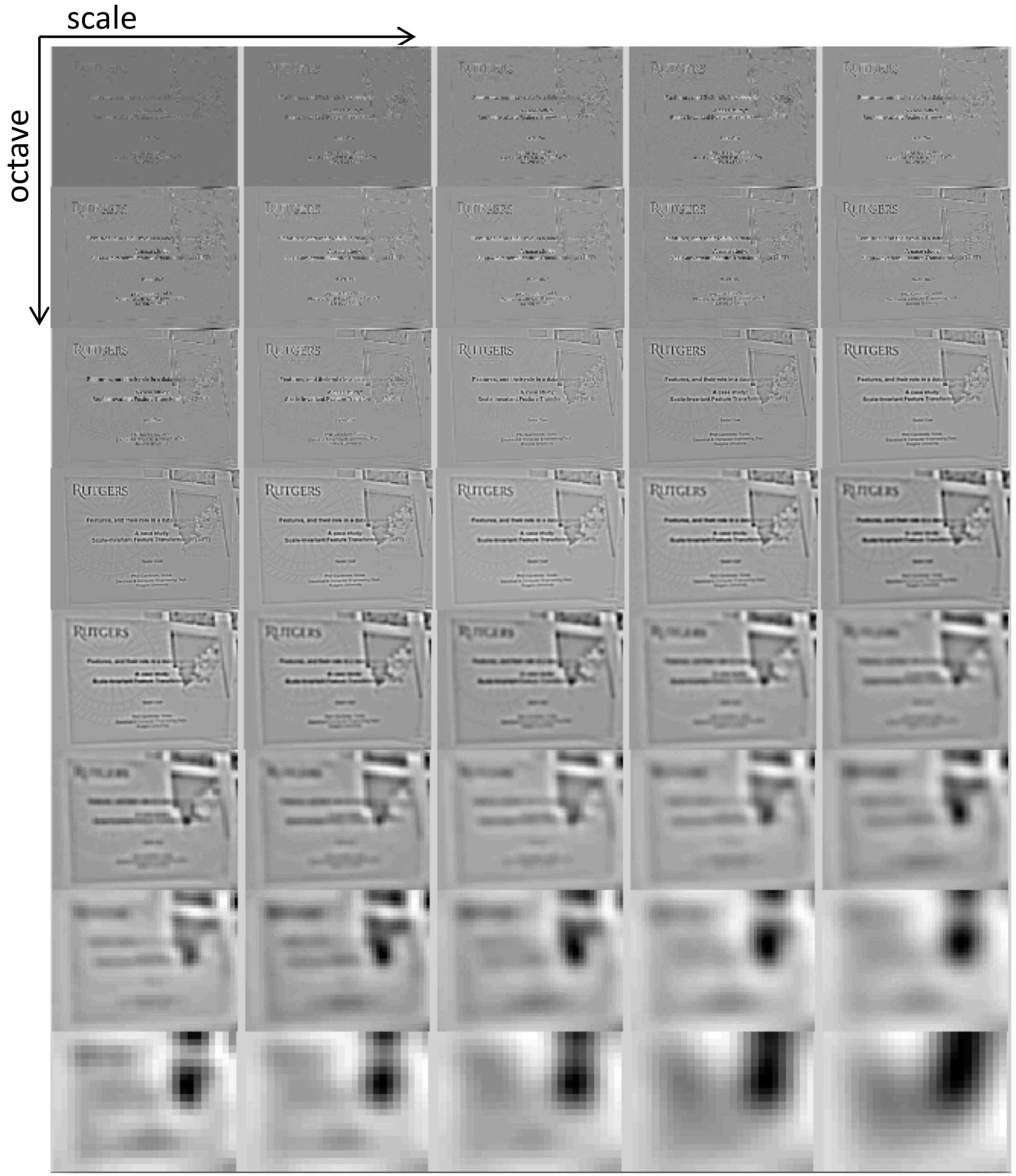
SIFT (1): Scale-space extrema detection- DoG Images



Original image



DoG Triplet



SIFT (2): Keypoint detection (post-processing)

Fine tuning of the location: Offset Computation

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma). \end{aligned}$$

$$\mathbf{x} = (x, y, \sigma)^T$$

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}.$$

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}}.$$

Check: $|D(\hat{\mathbf{x}})|$ >SomeOtherThreshold (such as 0.03)

Edge Removal

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\begin{aligned} \mathrm{Tr}(\mathbf{H}) &= D_{xx} + D_{yy} = \alpha + \beta, \\ \mathrm{Det}(\mathbf{H}) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \end{aligned}$$

Check if: $$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}.$$

Where $\alpha = r\beta$ (r = 10 in SIFT applications)

SIFT (3): Orientation Assignment

For “each keypoint” compute both gradient magnitude and orientation.

Gradient Magnitude: $m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$

Orientation: $\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$

Compute orientations and magnitude around the keypoint. And quantize the orientations into 36 bins (where 360 degrees are covered in 36 bins).

Keypoint: has a coordinate, scale, magnitude and the (maximum) orientation

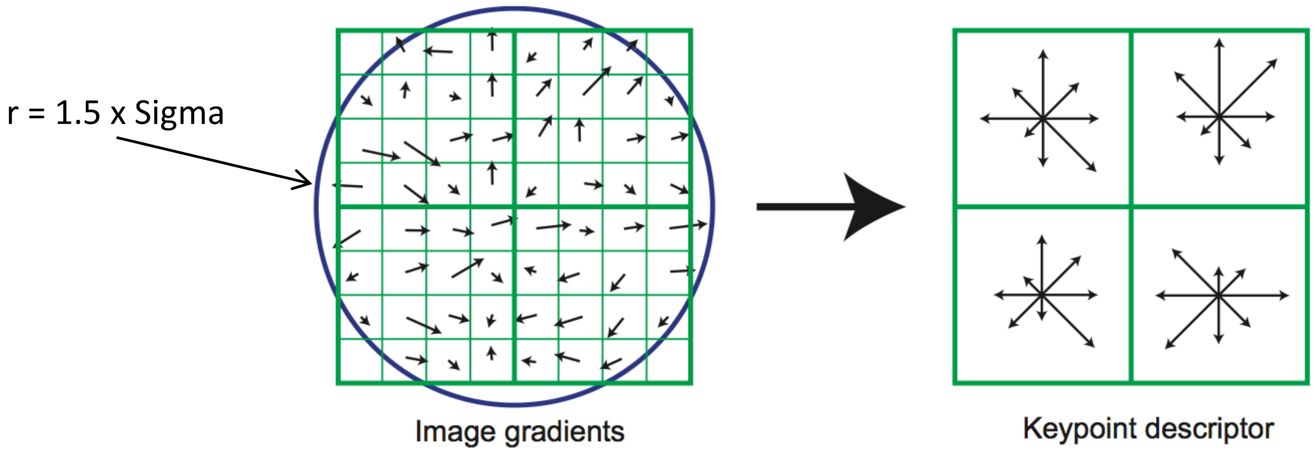
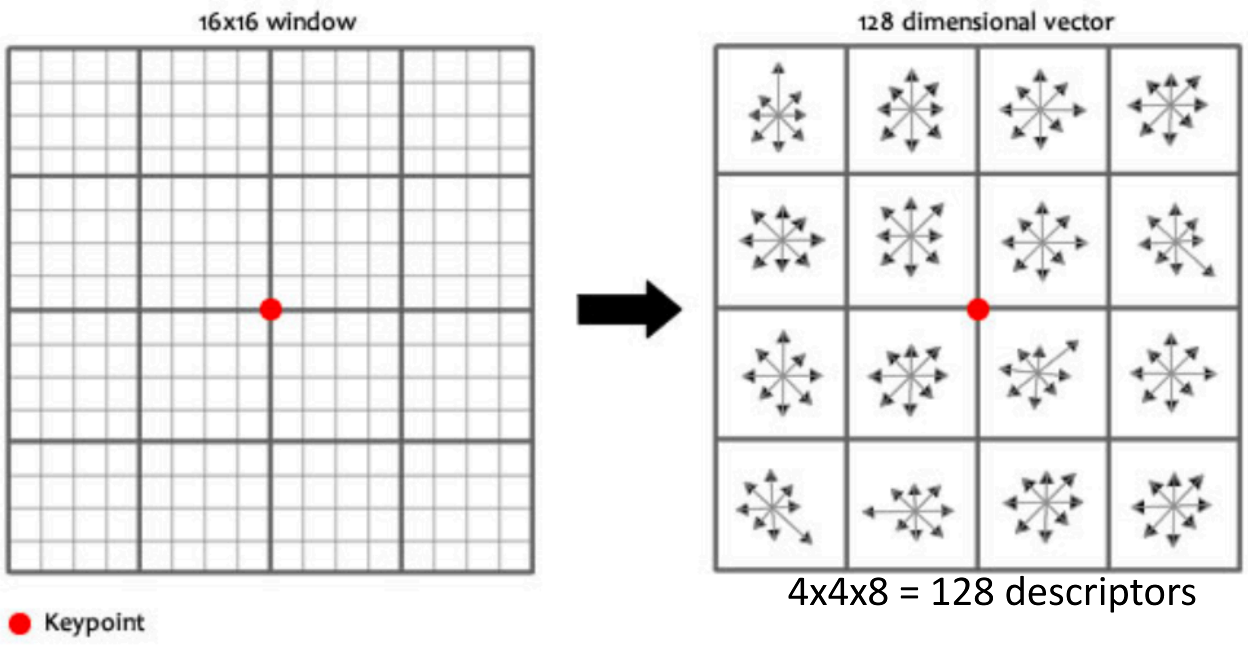
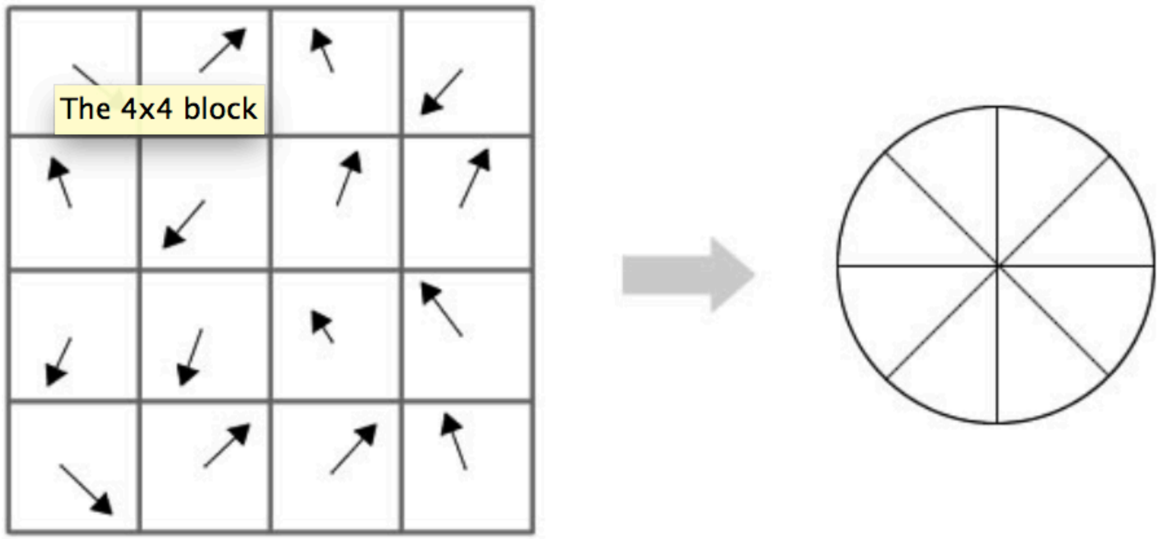


Image source: Lowe (2004)

SIFT (4): Orientation Assignment





Common SIFT applications

- Panorama stitching: (making a continuous one giant picture from parts),
- Human action recognition (e.g., detection of the movement of a human arm),
- Object recognition and retrieval (e.g., finding only the pictures including Sedat in a generic image database),
- Object tracking (the process of finding which object in the current time step is which one in the next time step).

Matching- recognition

- Do a similarity search,
  - Compute the ratio of the distances between the closest point and the next closest points.

- Cluster with Hough transform
  - Use x,y,magnitude and orientation (4 attributes) for clustering.

Sift – Results:

Image source: Lowe 1999

Sift – Results:

Image source: Lowe 1999

Sift Results: Image Stitching (2)

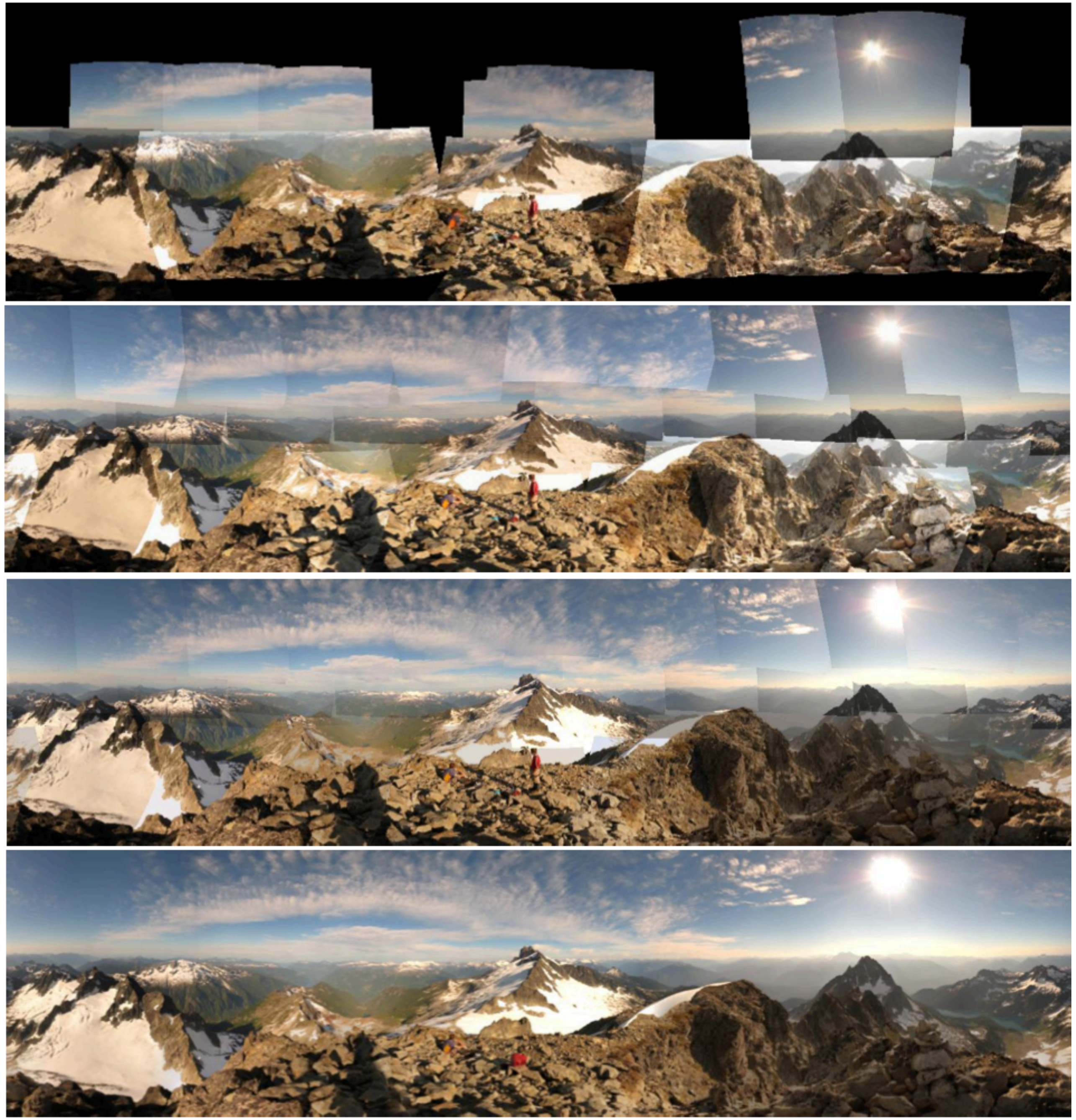
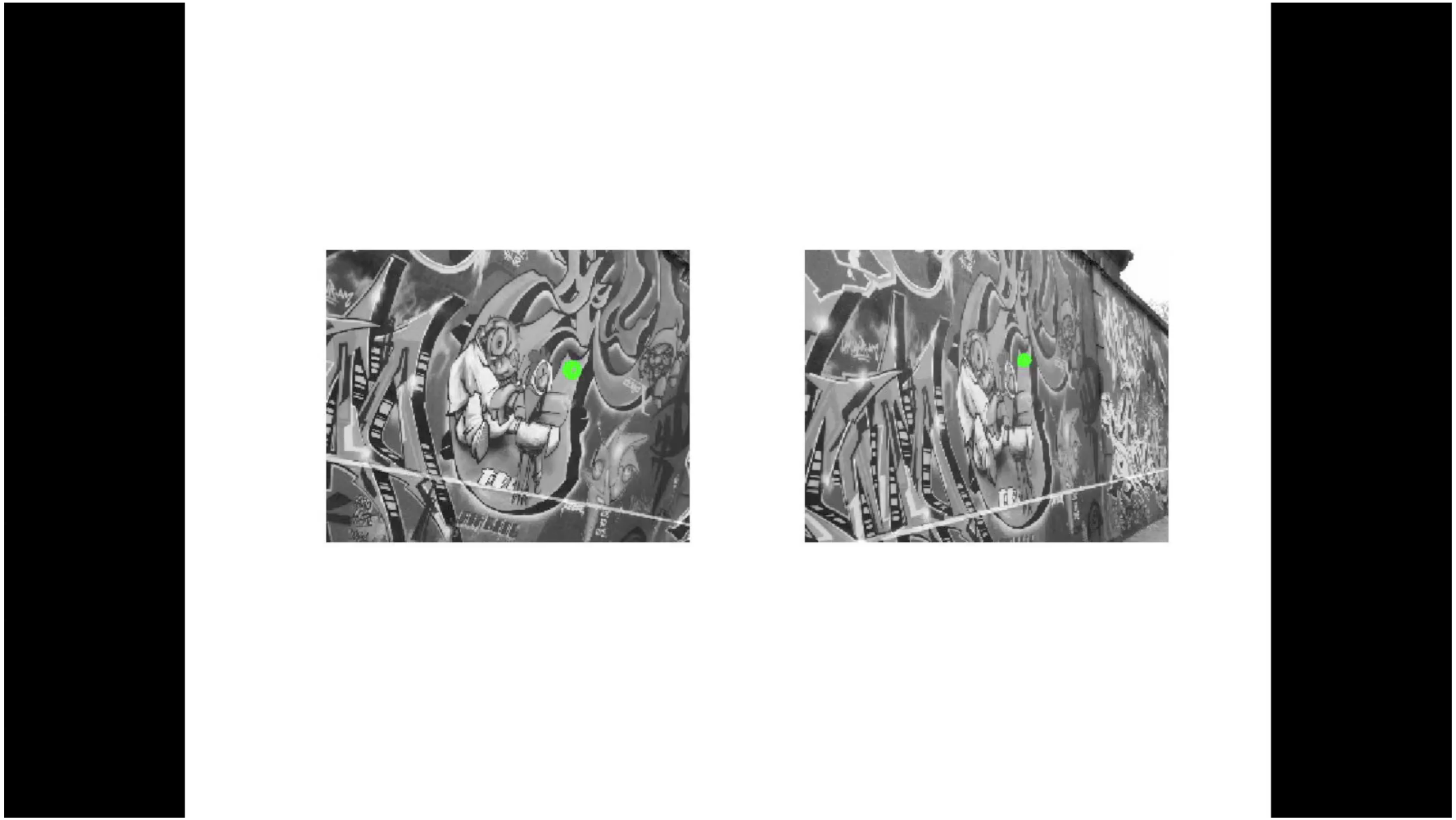


Image source: M. Brown, and D. G. Lowe. "Automatic panoramic image stitching using invariant features." *International Journal of Computer Vision* 74, no. 1 (2007): 59-73.

Feature matching

RANSAC is one of the common techniques!



Feature matching with SDM (this is not an easy task for SVM).
SIFT features are computed for both images and then used as input to the SDM.

SDM: Similarity Domains Machine (S. Ozer, 2018)

Video source: Sedat OZER

Examples of SIFT-like algorithms:

1) PCA-SIFT,
2) Color-SIFT,
3) Affine-SIFT,
4) SURF,

.

.

.

3D Sift

- SIFT is designed for 2D images. (data with only the x and y coordinates).

- What if we have 3D data (data with x,y,z coordinates) ?

3D Sift

- SIFT is designed for 2D images. (data with only the x and y coordinates).

What if we have the depth information as well (i.e., the data comes with x, y and z coordinates)?

3D Sift

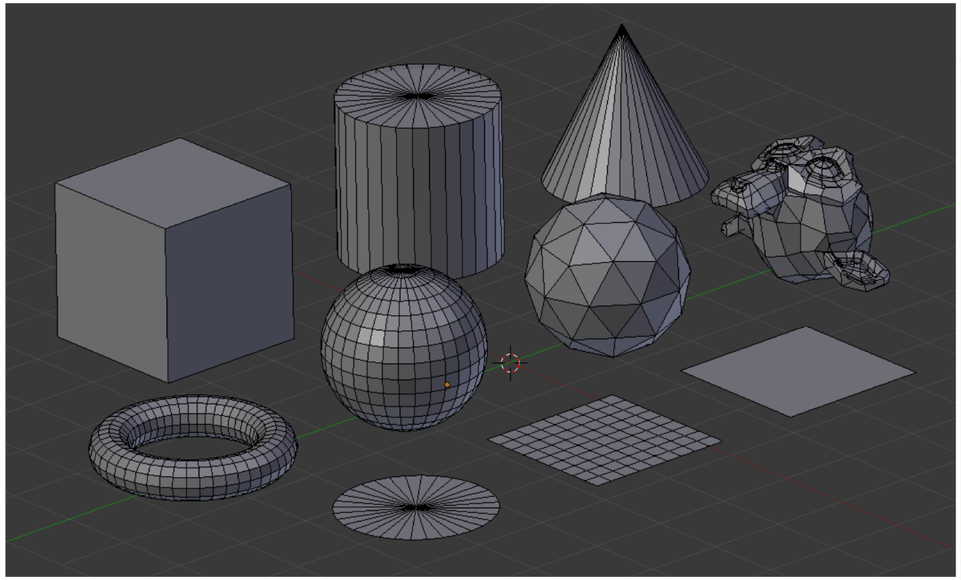
- A generalization of 2D sift onto 3D datasets.

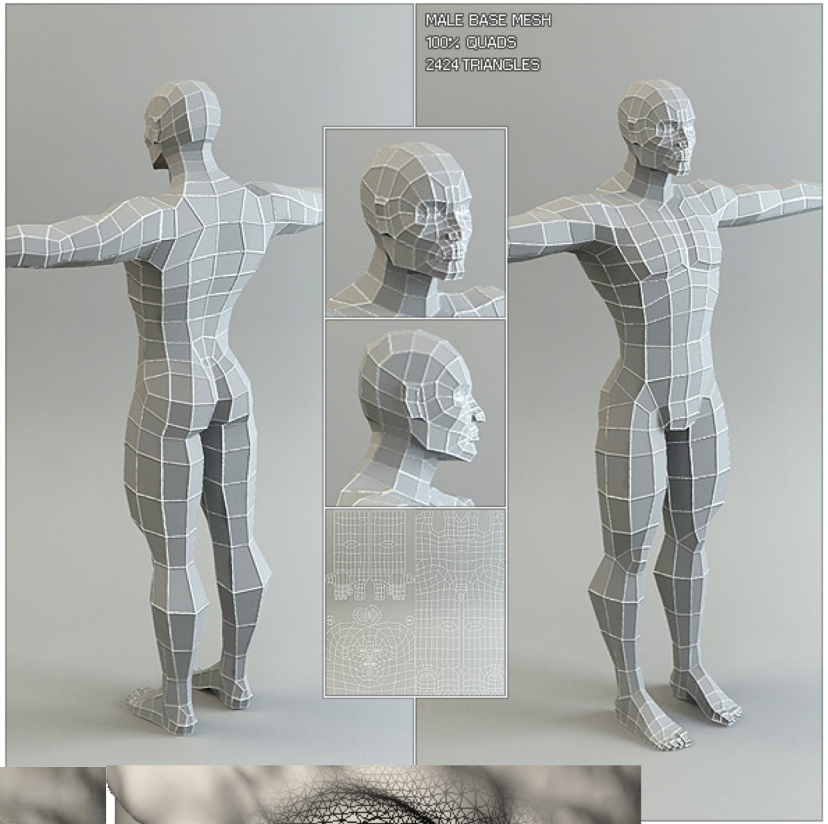
Scovanner, Ali & Shah, 2007 (from **UCF**)

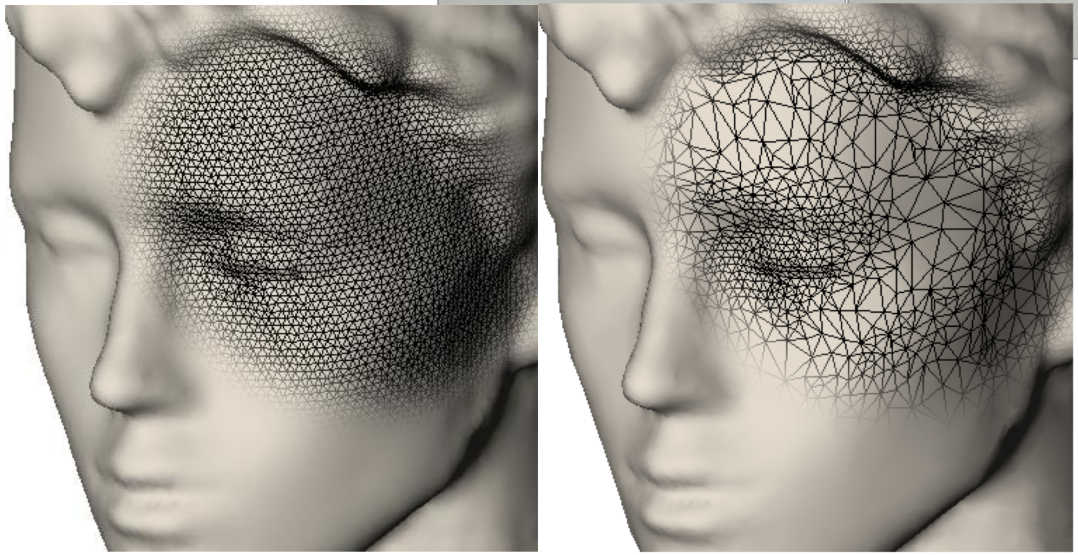
- Use **3D Sift** in:
  - activity detection and activity recognition
  - medical imaging applications
  - …

- If saving data as 3D is too complicated and almost impossible:
  - use an abstraction for 3D data: use mesh
    - Mesh-Sift

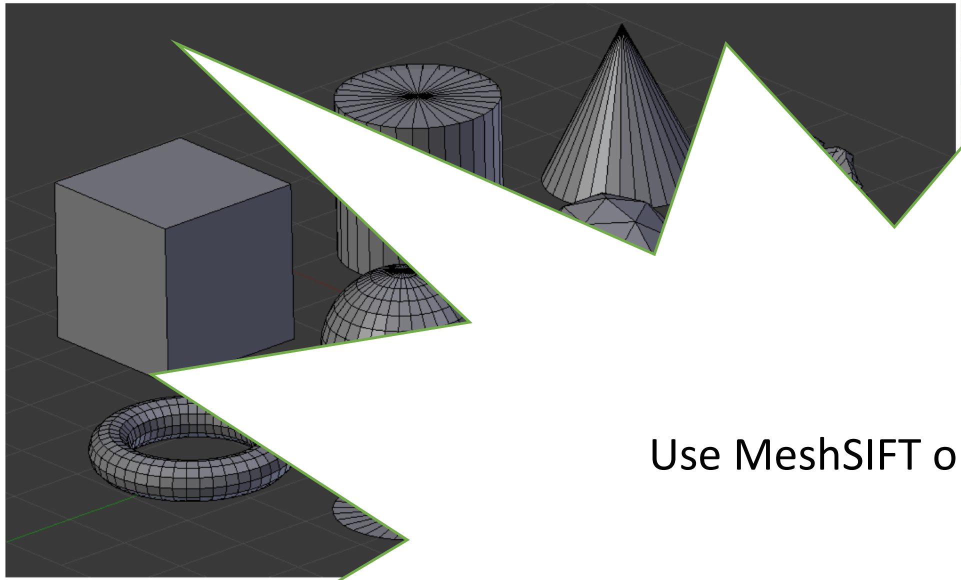
Mesh:

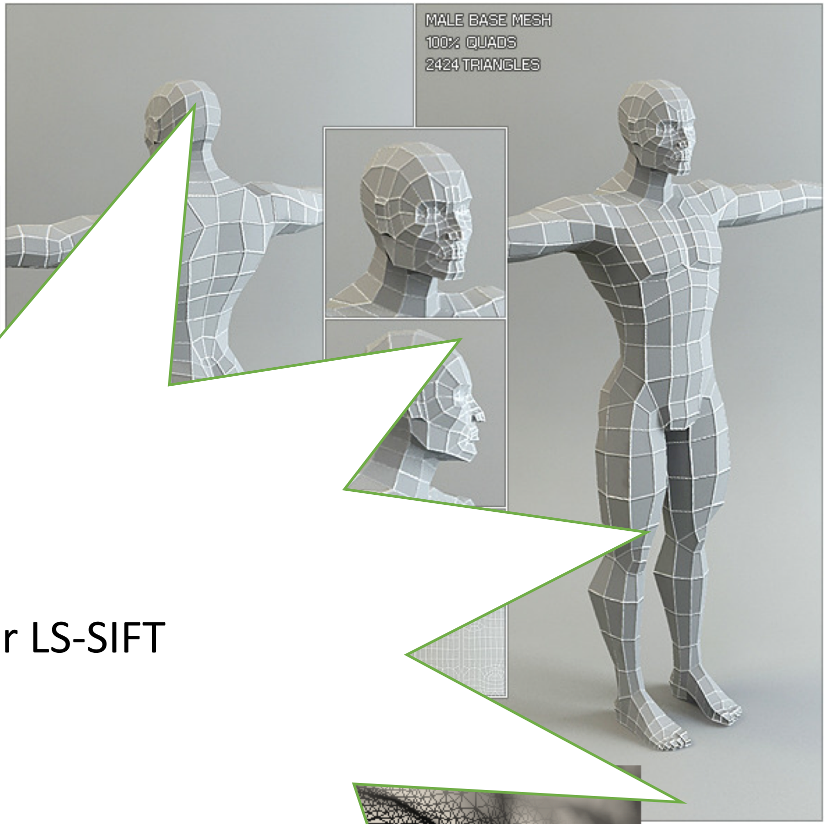


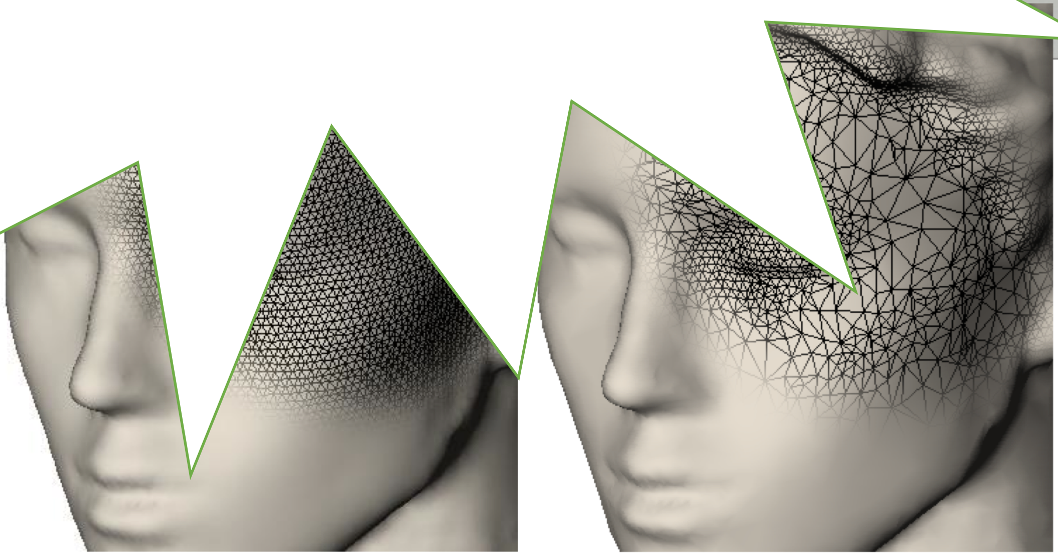




Mesh:







Use MeshSIFT or LS-SIFT

Summary

- SIFT is patented! Legally, be careful to use it in industrial applications,
- Many SIFT-like algorithms have been derived following the original SIFT algorithm's steps,
- SIFT-like algorithms have been used in many image-based applications.
- Many similarities between the common feature types: HOG, SIFT and SURF algorithms!

Questions?

Note: Most of these slides were prepared by Sedat Ozer during his Ph.D studies at Rutgers.
Also included some content from Ali Farhadi (from Univ. of Washington)

Additional Resource about RANSAC

- https://people.cs.umass.edu/~elm/Teaching/ppt/370/370_10_RANSAC.pptx.pdf