Scale Invariant Feature Transform (SIFT)

CSE 6367: Computer Vision

Instructor: William J. Beksi

Introduction

- The **Scale Invariant Feature Transform** (SIFT) is a feature detection algorithm created by David Lowe (2004) and patented by the University of British Columbia

Introduction

- The **Scale Invariant Feature Transform** (SIFT) is a feature detection algorithm created by David Lowe (2004) and patented by the University of British Columbia
- SIFT extracts distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene

Introduction

- The **Scale Invariant Feature Transform** (SIFT) is a feature detection algorithm created by David Lowe (2004) and patented by the University of British Columbia
- SIFT extracts distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene
- Applications include object recognition where SIFT can robustly identify objects among clutter and occlusion while achieving near real-time performance

Prior Work

- The development of image matching using a set of local interest points can be traced back to Moravec (1981) on stereo matching using a corner detector

Prior Work

- The development of image matching using a set of local interest points can be traced back to Moravec (1981) on stereo matching using a corner detector
- The Moravec detector was improved by Harris and Stephens (1988) to make it more repeatable under small image variations and near edges

Prior Work

- The development of image matching using a set of local interest points can be traced back to Moravec (1981) on stereo matching using a corner detector
- The Moravec detector was improved by Harris and Stephens (1988) to make it more repeatable under small image variations and near edges
- The Harris corner detector which selects any image location that has large gradients in all directions at a predetermined scale, not just corners, had been widely used for image matching tasks

Prior Work

- The groundbreaking work of Schmid and Mohr (1997) showed that invariant local feature matching could be extended to general image recognition problems where a feature is matched against a large database of images

Prior Work

- The groundbreaking work of Schmid and Mohr (1997) showed that invariant local feature matching could be extended to general image recognition problems where a feature is matched against a large database of images
- They used Harris corners to select interest points, but rather than matching with a correlation window they used a rotationally invariant descriptor of the local image region

Prior Work

- The groundbreaking work of Schmid and Mohr (1997) showed that invariant local feature matching could be extended to general image recognition problems where a feature is matched against a large database of images
- They used Harris corners to select interest points, but rather than matching with a correlation window they used a rotationally invariant descriptor of the local image region
- This allowed features to be matched under arbitrary orientation change between two images thus enabling general object recognition

Steps for Extracting Keypoints

1. **Scale-space extrema detection** - identify potential locations for finding features using the difference of Gaussians (DoG)

Steps for Extracting Keypoints

1. **Scale-space extrema detection** - identify potential locations for finding features using the difference of Gaussians (DoG)
2. **Keypoint localization** - accurately locate the feature keypoints based on their stability

Steps for Extracting Keypoints

1. **Scale-space extrema detection** - identify potential locations for finding features using the difference of Gaussians (DoG)
2. **Keypoint localization** - accurately locate the feature keypoints based on their stability
3. **Orientation assignment** - assign orientation to keypoints based on local image gradients

Steps for Extracting Keypoints

1. **Scale-space extrema detection** - identify potential locations for finding features using the difference of Gaussians (DoG)
2. **Keypoint localization** - accurately locate the feature keypoints based on their stability
3. **Orientation assignment** - assign orientation to keypoints based on local image gradients
4. **Keypoint descriptor** - transform the keypoint into a high-dimensional vector representation

Scales

- What should be the value of the standard deviation of the Gaussian distribution, σ, when doing edge detection?

Scales

- What should be the value of the standard deviation of the Gaussian distribution, σ, when doing edge detection?
- If we use multiple σ values, then how do we combine multiple edge maps?

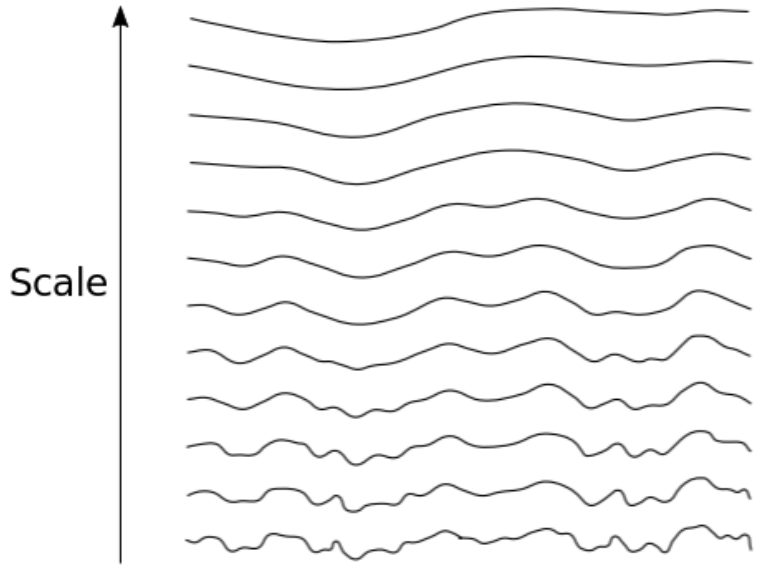
Scale-Space

- We can apply whole spectrum of scales (Witkin, 1983)

Scale-Space

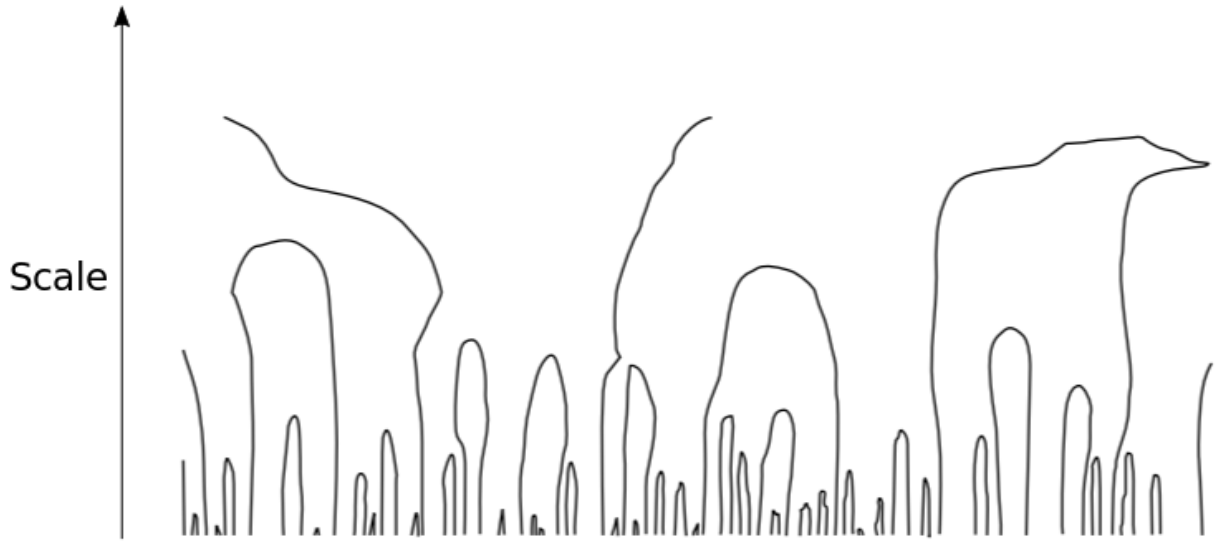
- We can apply whole spectrum of scales (Witkin, 1983)
- Plot the **zero-crossings** (i.e. where the edges are in the Laplacian of Gaussian) versus the scales in a scale-space

Scale-Space Filtering



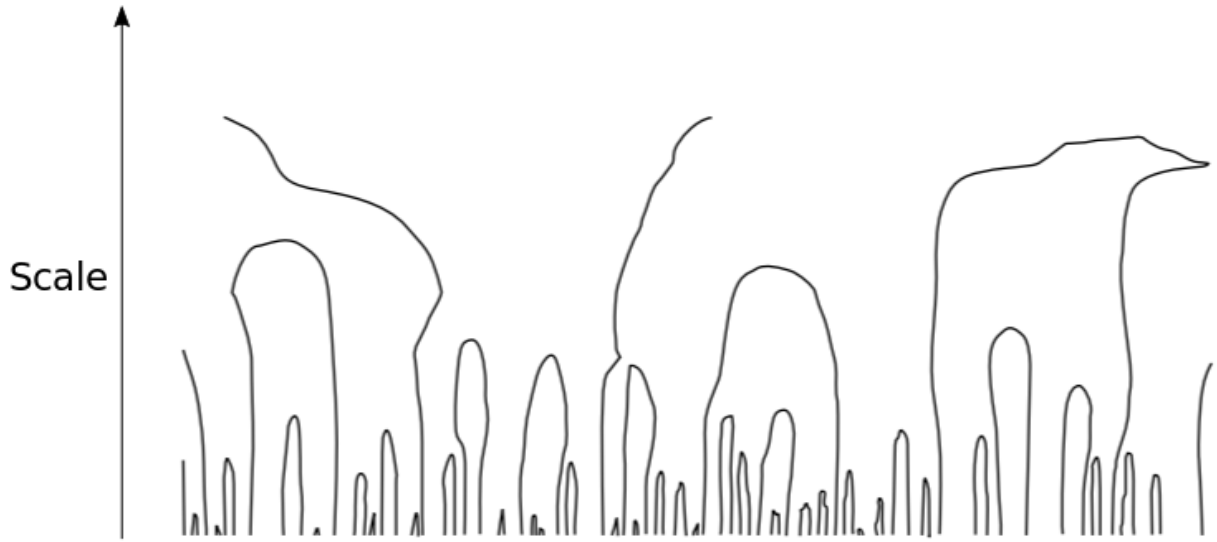
- Multiple smooth versions of a 1D signal (i.e. a single row of an image)

Scale-Space Filtering



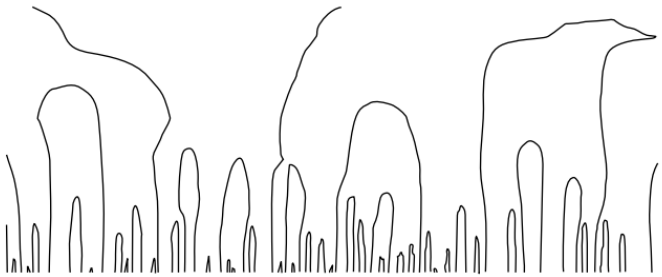
- At the lower scales we have many zero-crossings

Scale-Space Filtering

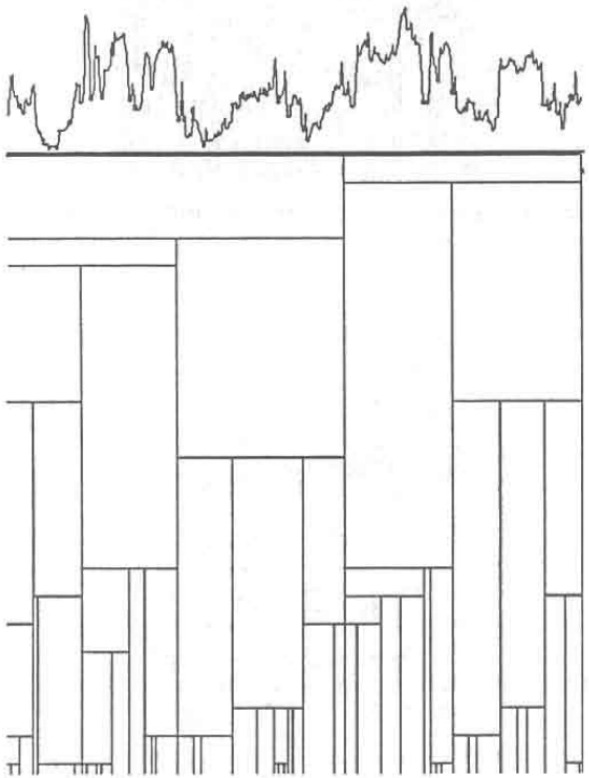


- At the lower scales we have many zero-crossings
- As we increase the scale the zero-crossings disappear (merge)

Scale-Space Filtering



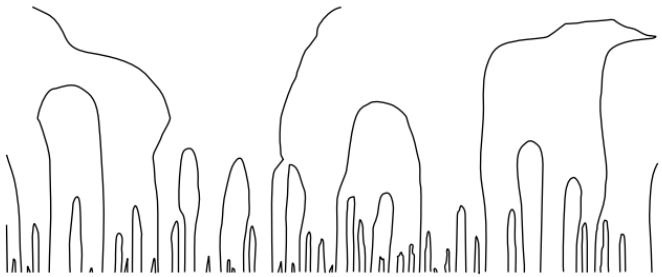
Scale Space



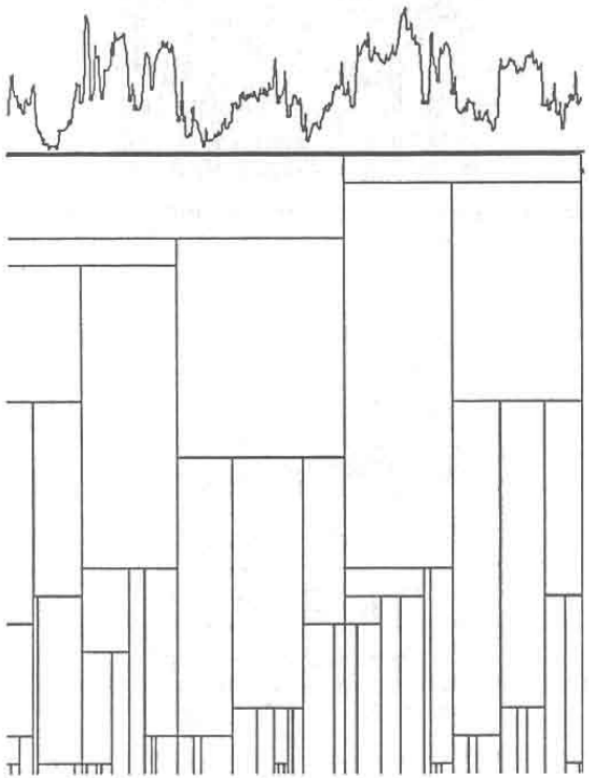
Interval Tree

- We can encode the zero-crossings in an **interval tree**

Scale-Space Filtering



Scale Space



Interval Tree

- We can encode the zero-crossings in an **interval tree**
- The interval tree provides insight on the stability of the zero-crossings

Scale-Space Filtering

- Within the scale-space, we interpret the contours as arches that are open at the bottom and closed at the top

Scale-Space Filtering

- Within the scale-space, we interpret the contours as arches that are open at the bottom and closed at the top
- In the interval tree, each interval corresponds to a node in the tree whose parent node represents a merged interval and whose children represent smaller intervals

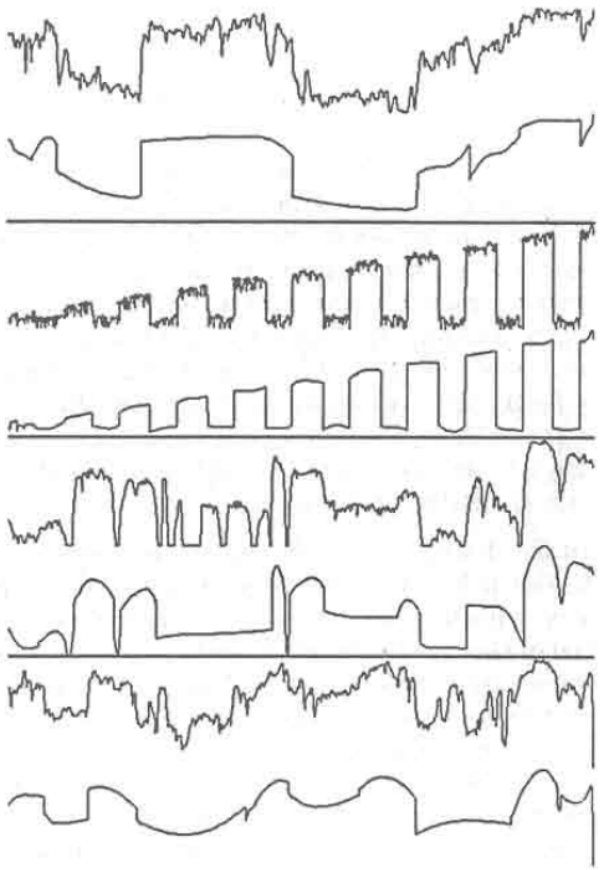
Scale-Space Filtering

- The stability of a node is represented as a scale range over which the interval exists in the interval tree

Scale-Space Filtering

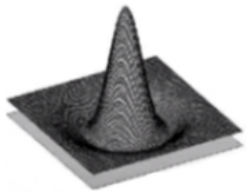
- The stability of a node is represented as a scale range over which the interval exists in the interval tree
- We can obtain a top-level description of the scale-space by iteratively removing nodes from the interval tree (i.e. splice out nodes that are less stable than their parents)

Scale-Space Filtering

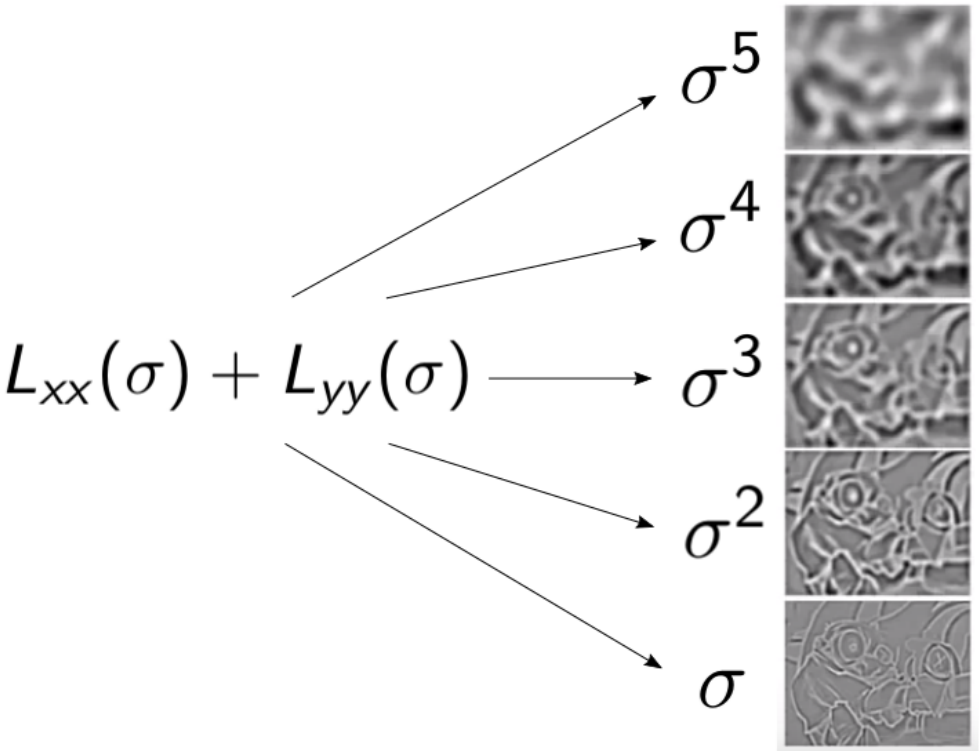


- A top-level description of several signals using the stability criterion

Laplacian of Gaussian (LoG)

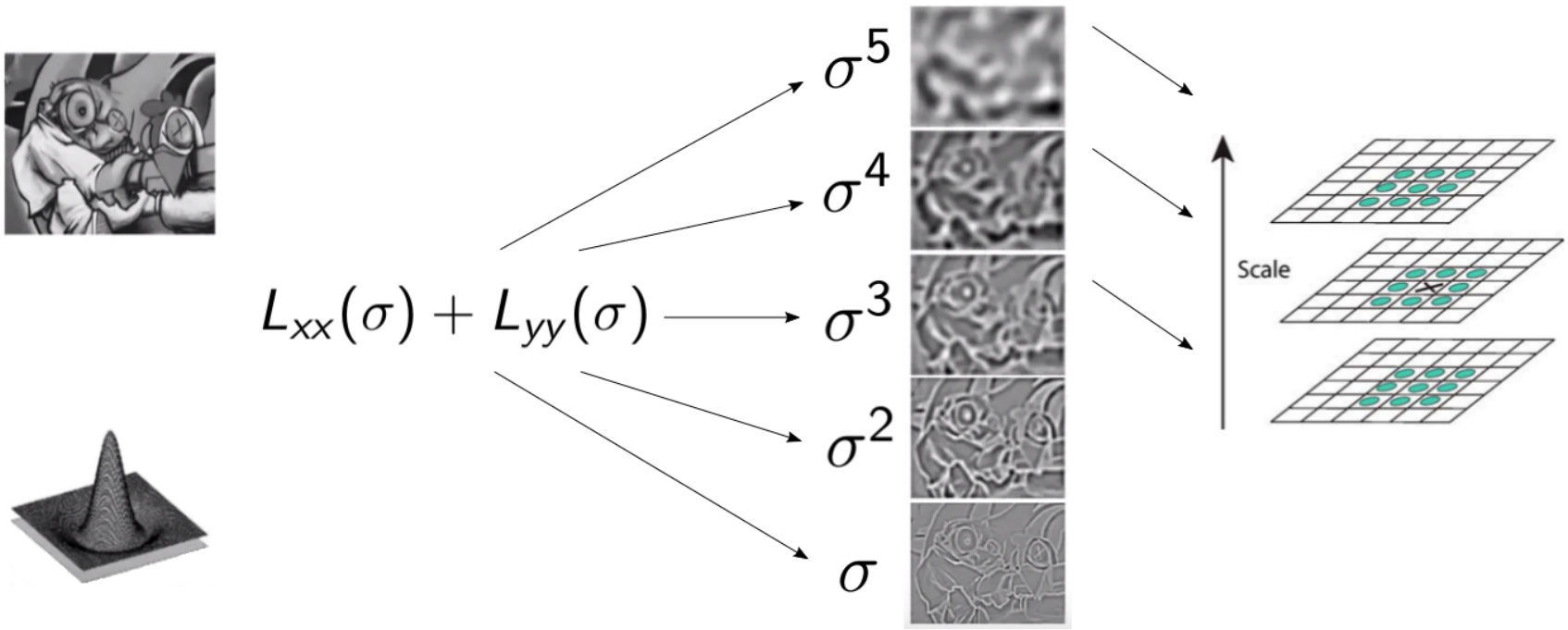


$$L_{xx}(\sigma) + L_{yy}(\sigma)$$



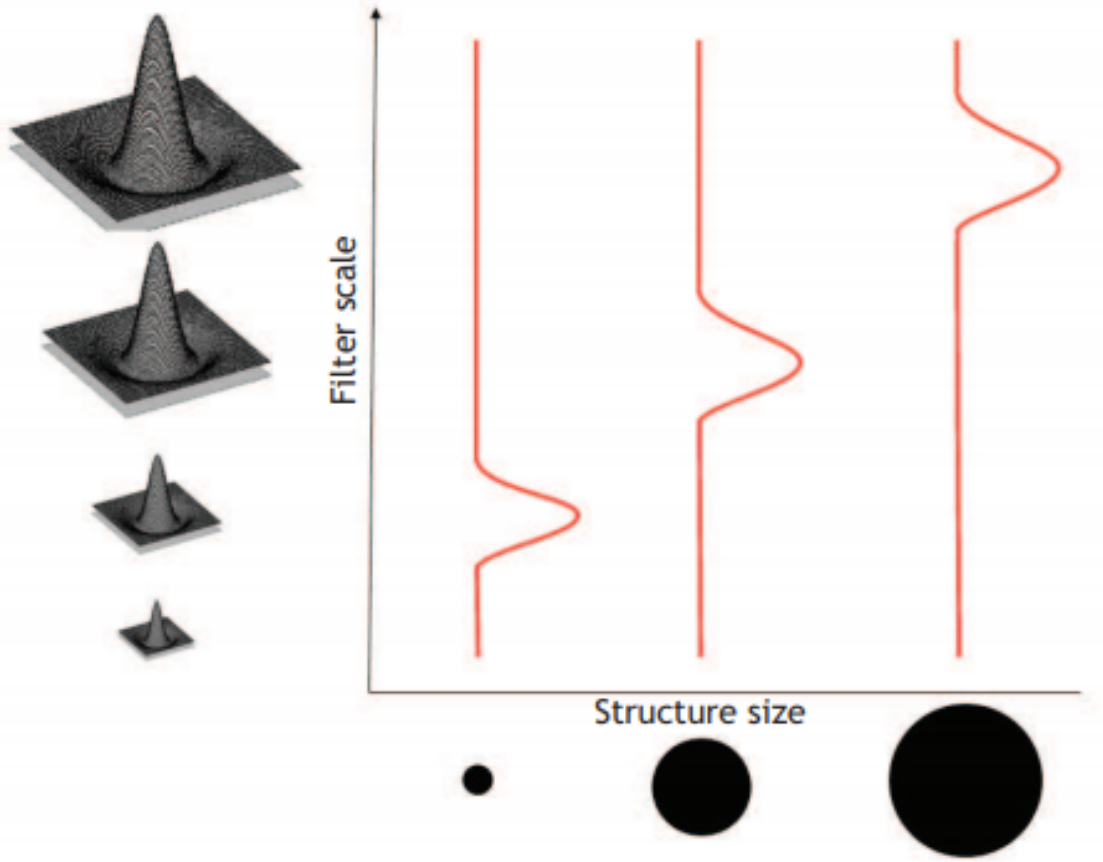
- The interest points are the local maxima in the scale-space of the Laplacian of Gaussian (LoG)

Laplacian of Gaussian (LoG)



- If the center point (X) is an extrema (minima or maxima) of the 26 points, then it is a potential interest point

Laplacian of Gaussian (LoG)



- The LoG can be used as a **blob detector**

Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

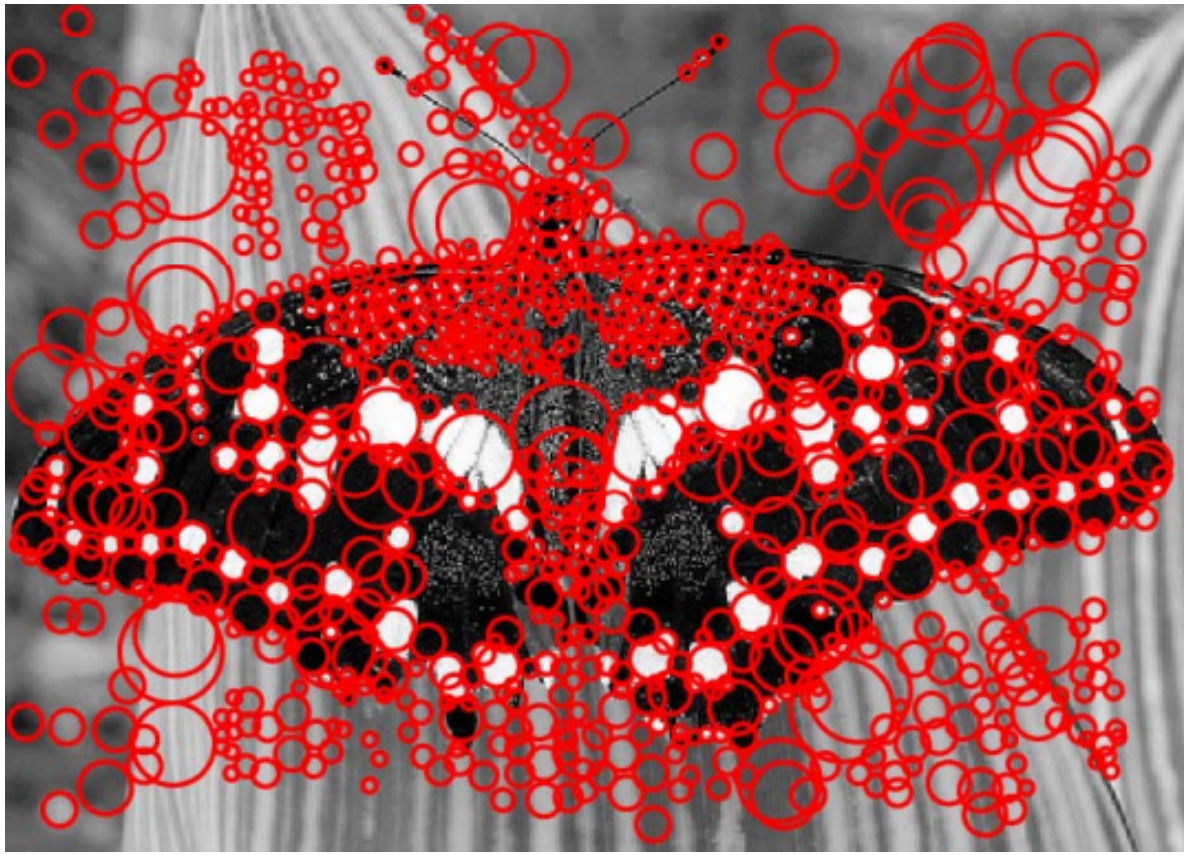
Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

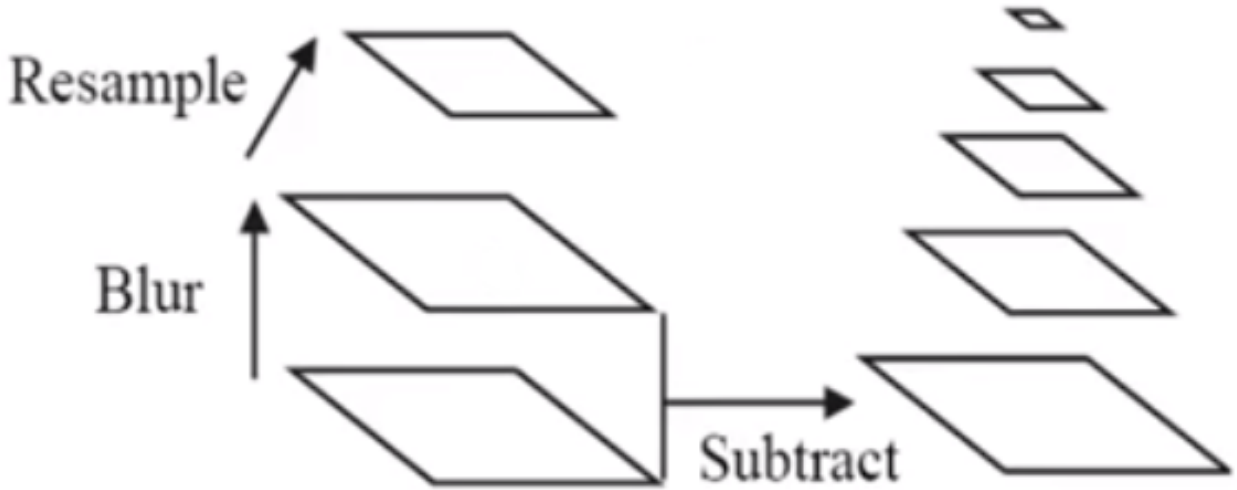
Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

Example: Scale-Space Blob Detector

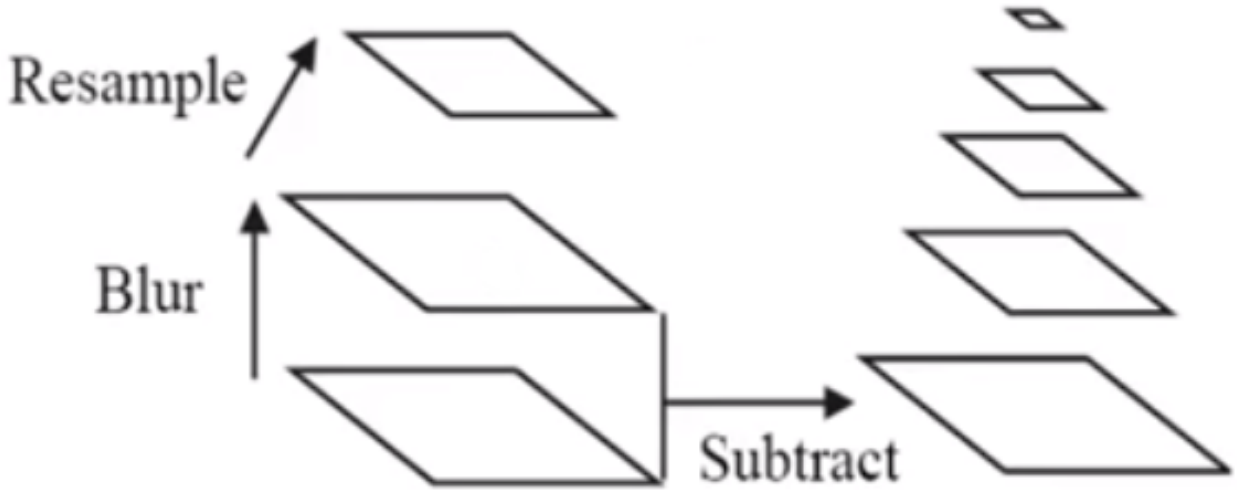


Building a Scale-Space



- We must examine all scales to identify scale invariant features

Building a Scale-Space



- We must examine all scales to identify scale invariant features
- An efficient way to do this is to compute the Laplacian pyramid using DoG (Burt and Adelson, 1983)

Approximation of LoG by DoG

- From the heat diffusion equation $\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G$, we can compute $\nabla^2 G$ from the finite difference approximation to $\partial G / \partial \sigma$ using the difference of nearby scales at $k\sigma$ and σ

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

and therefore

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

where

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

Approximation of LoG by DoG

- The approximation error goes to zero as k goes to 1

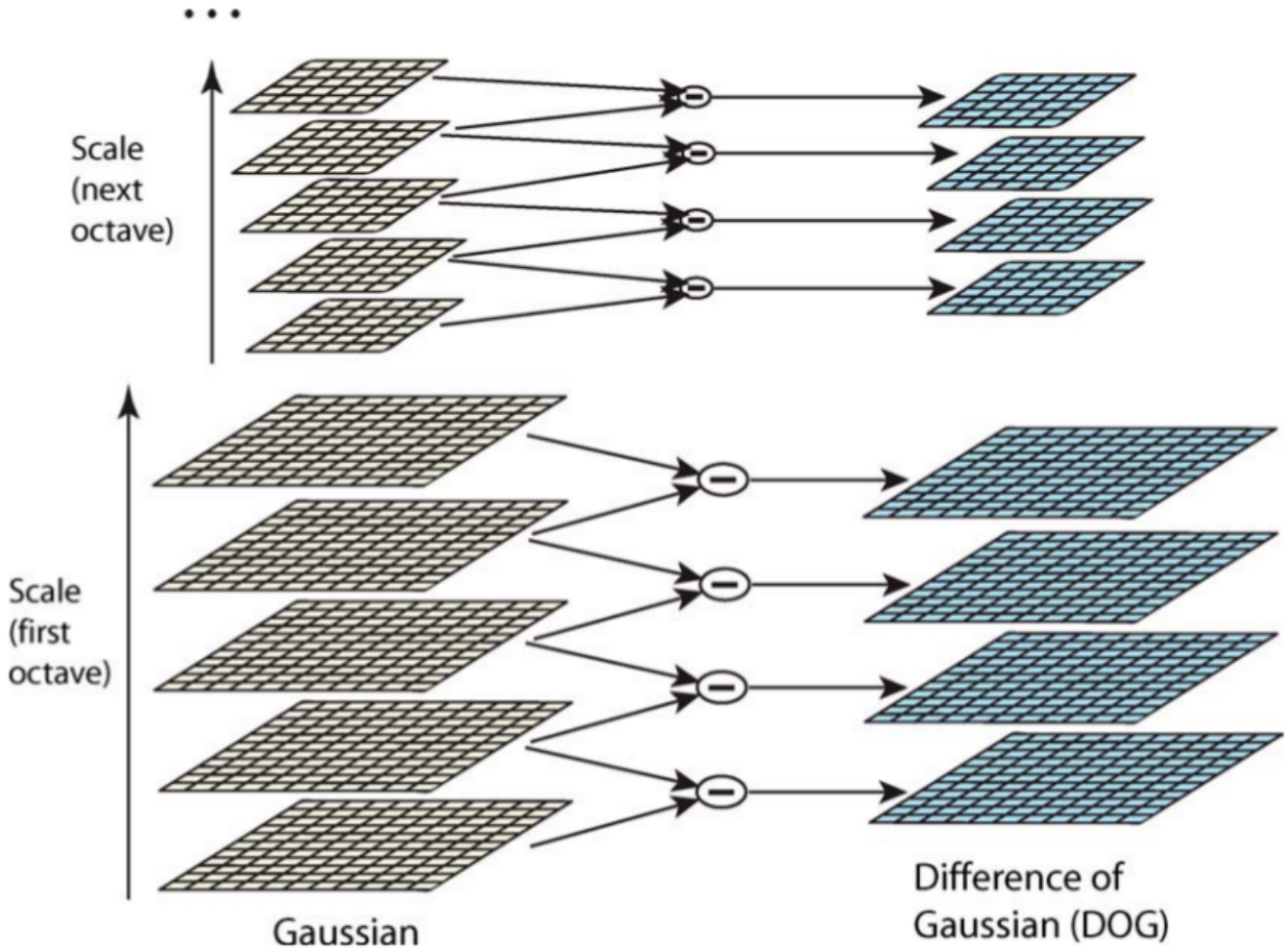
Approximation of LoG by DoG

- The approximation error goes to zero as k goes to 1
- In practice, the approximation has almost no impact on the stability of extrema detection or localization even for significant differences in scale

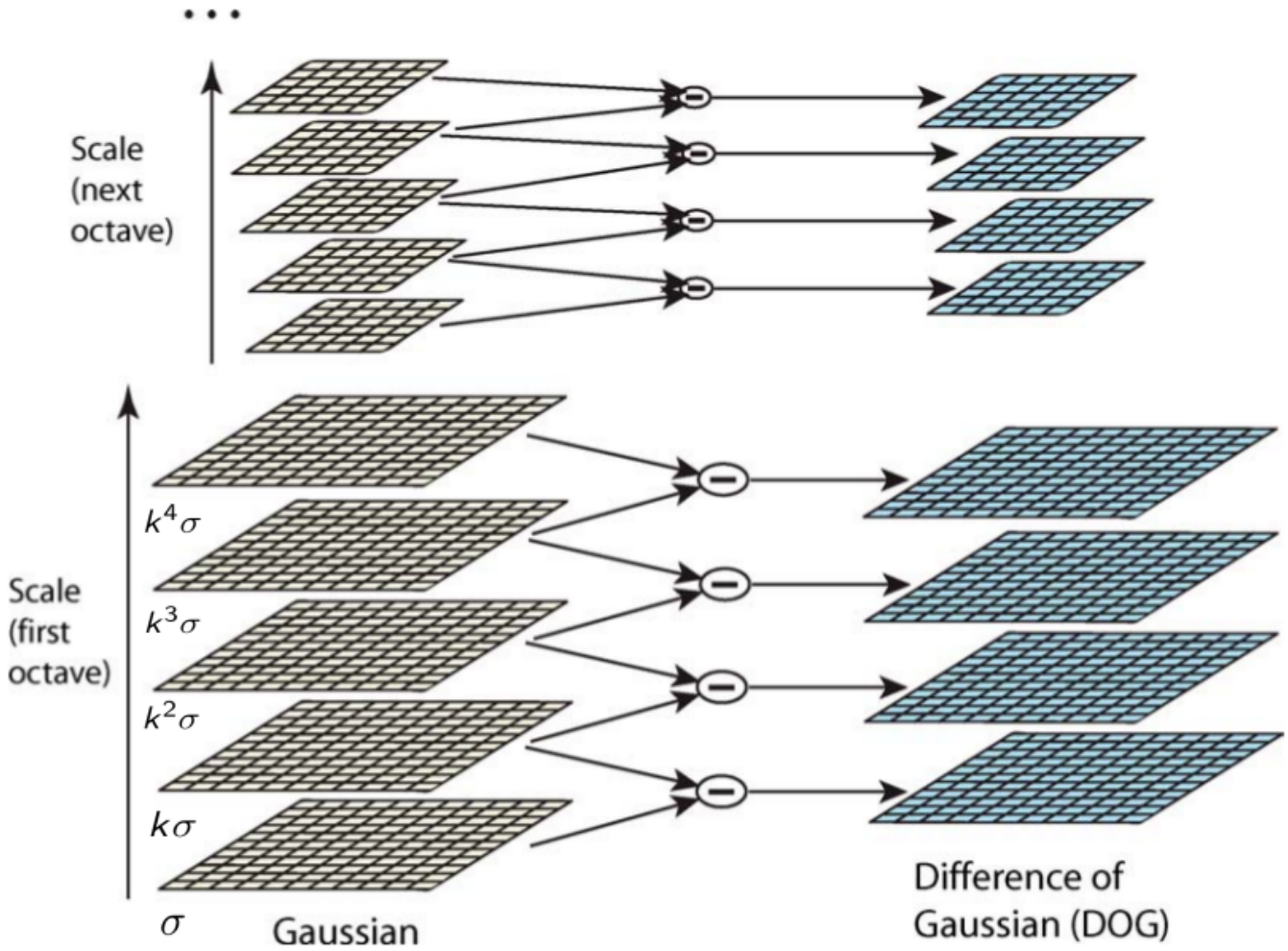
Approximation of LoG by DoG

- The approximation error goes to zero as k goes to 1
- In practice, the approximation has almost no impact on the stability of extrema detection or localization even for significant differences in scale
- Typical values are $\sigma = 1.6$, $k = \sqrt{2}$

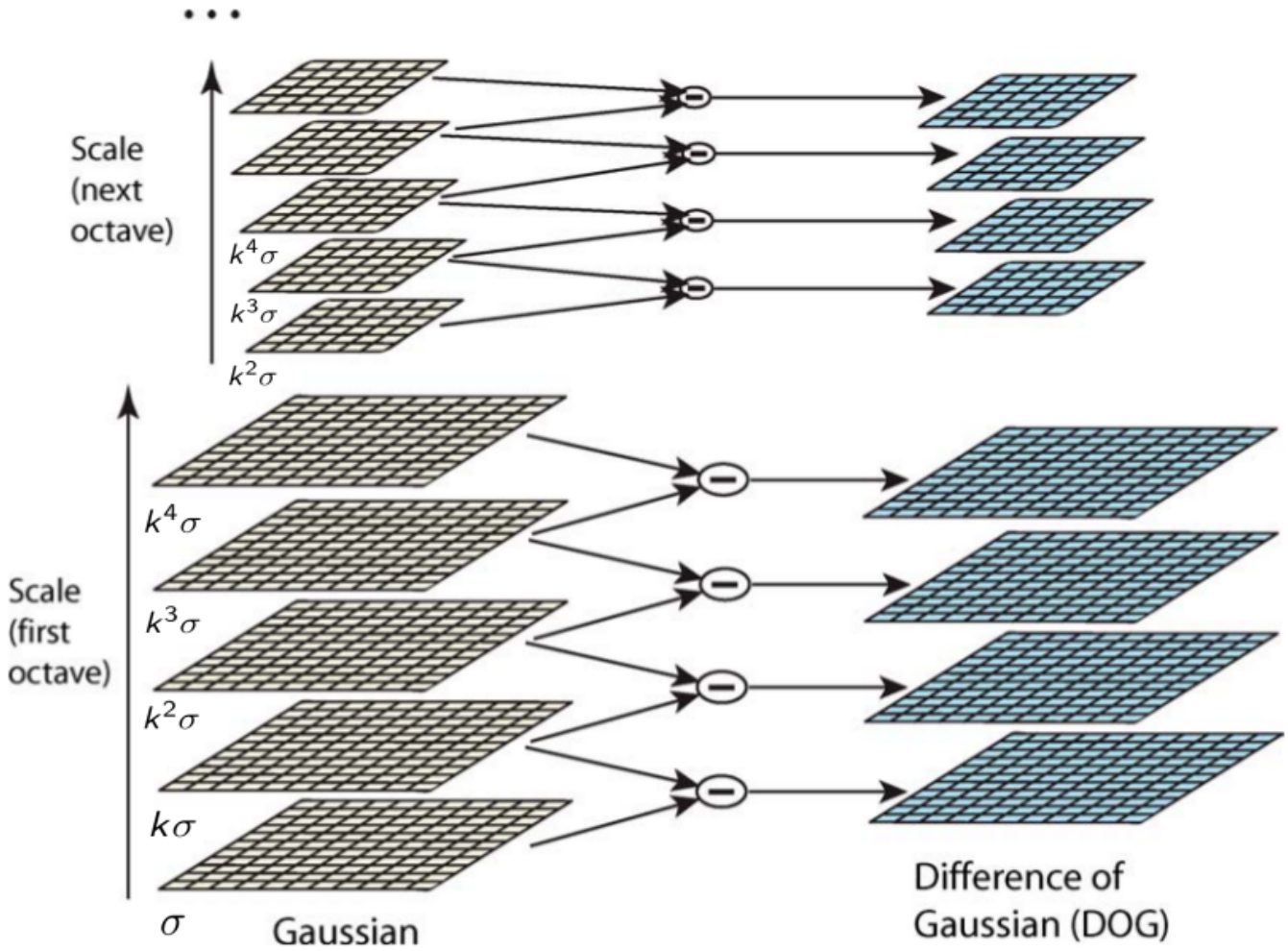
Building a Scale-Space



Building a Scale-Space

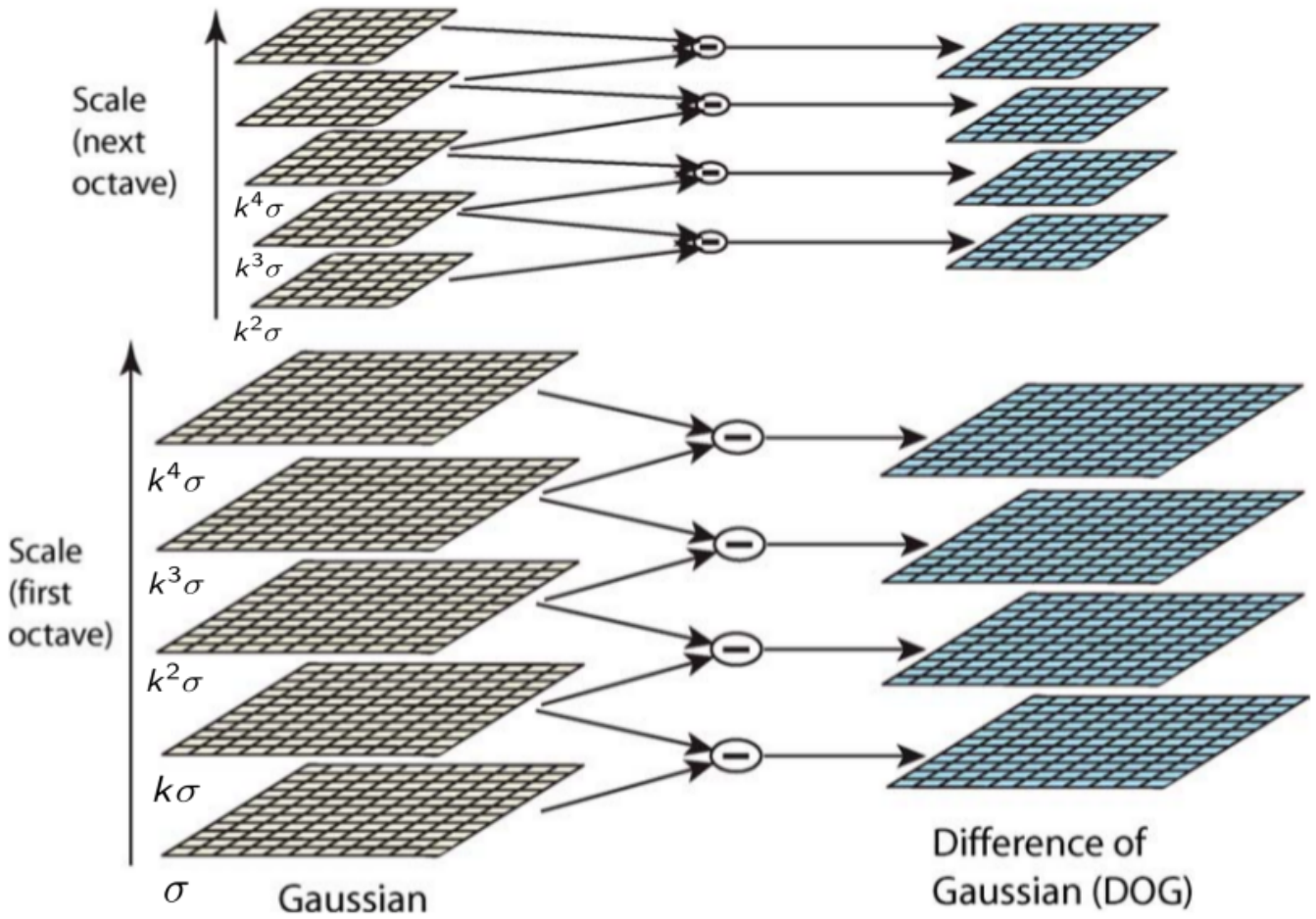


Building a Scale-Space



Building a Scale-Space

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

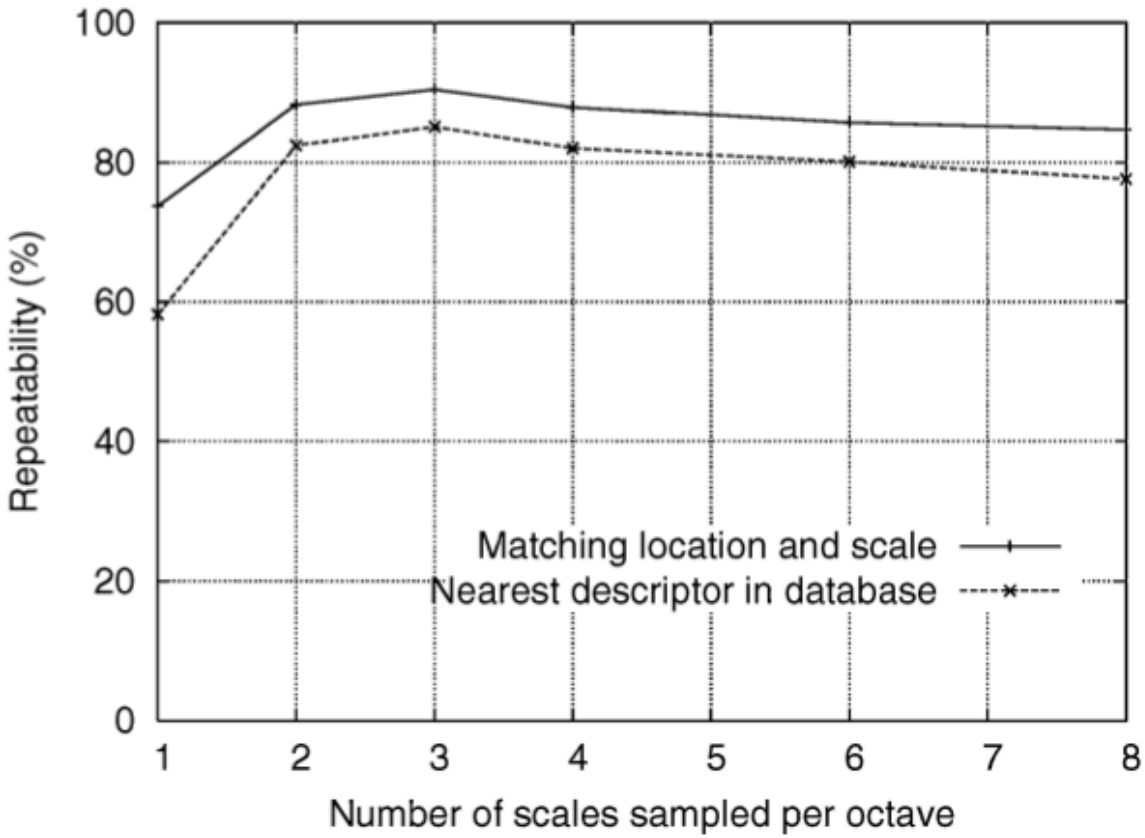


Building a Scale-Space

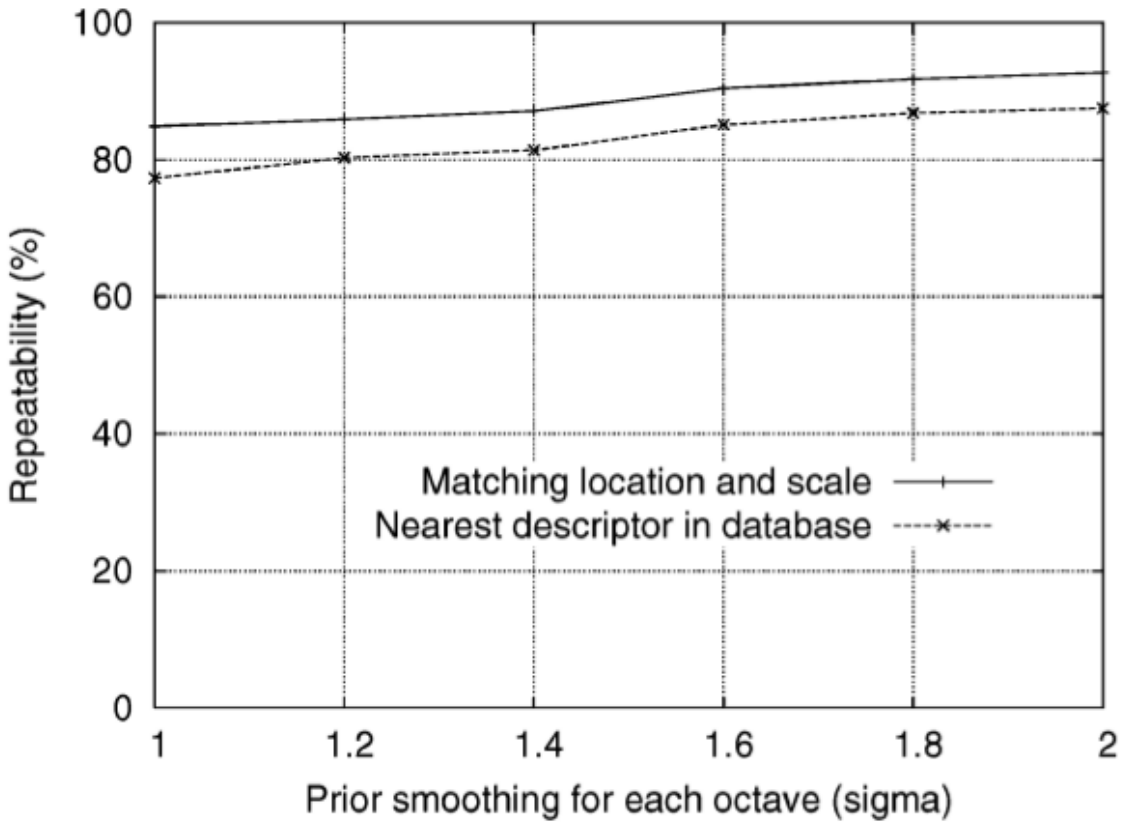
	scale →				
octave	0.707107	1.000000	1.414214	2.000000	2.828427
	1.414214	2.000000	2.828427	4.000000	5.656854
	2.828427	4.000000	5.656854	8.000000	11.313708
	5.656854	8.000000	11.313708	16.000000	22.627417

- Empirical results show that these scales comprehensively cover all images

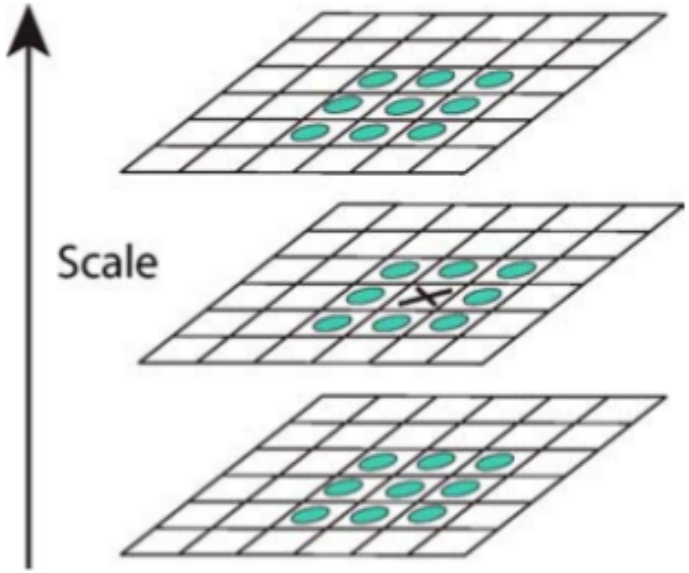
How Many Scales per Octave?



Initial Value of Sigma

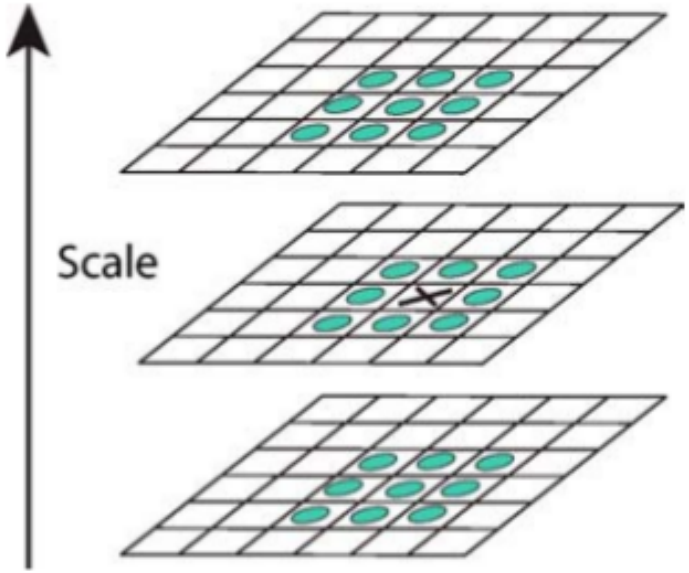


Scale-Space Peak Detection



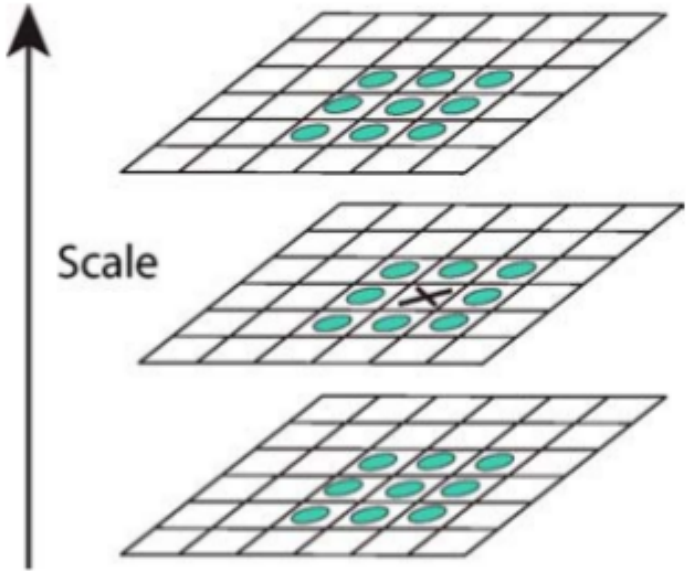
- We compare a pixel (X) with 26 neighboring pixels (green circles) in the current and adjacent scales

Scale-Space Peak Detection



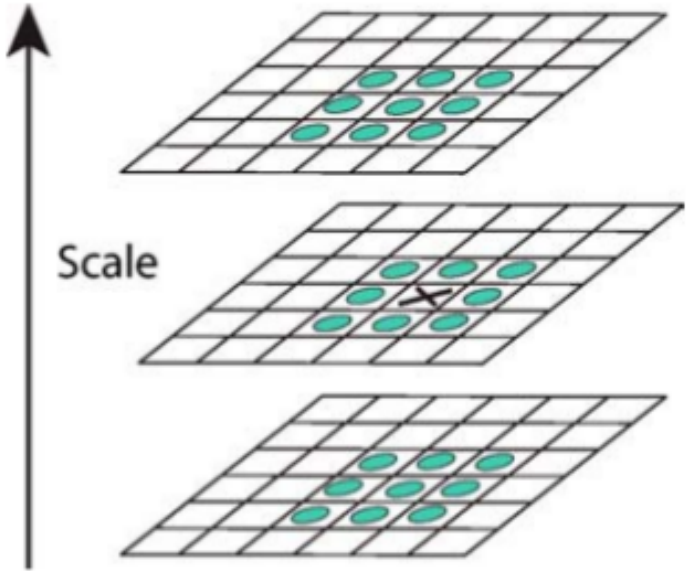
- We compare a pixel (X) with 26 neighboring pixels (green circles) in the current and adjacent scales
- The pixel (X) is selected if it is larger/smaller than its neighbors

Scale-Space Peak Detection



- A large number of extrema can make the detection computationally expensive

Scale-Space Peak Detection



- A large number of extrema can make the detection computationally expensive
- Therefore, we want to detect the most stable subset with a coarse sampling of scales

Locating Keypoints

(a) Original image

(b) Extrema locations

- Given an image (a), keypoint candidates are chosen by extrema detection (b)

Initial Outlier Rejection

- We want to remove keypoint candidates that have low contrast or are poorly localized along an edge

Initial Outlier Rejection

- We want to remove keypoint candidates that have low contrast or are poorly localized along an edge
- To perform the initial outlier rejection, we'll use the Taylor series expansion of the scale-space function (i.e. DoG), D,

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

where D and its derivatives are evaluated at the sample point and $\mathbf{x} = [x, y, \sigma]^T$ is the offset from this point

Initial Outlier Rejection

- The location of extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of D w.r.t $\mathbf{x}$ and setting it to zero

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

Initial Outlier Rejection

- The location of extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of D w.r.t $\mathbf{x}$ and setting it to zero

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

- The value of $D(\mathbf{x})$ at a minima or maxima must be large, i.e. $|D(\mathbf{x})| >$ threshold

Initial Outlier Rejection

(a) 832 keypoints

(b) 729 keypoints

- The number of keypoints is reduced from 832 (a) to 729 (b) using a threshold of 0.03

Further Outlier Rejection

- The DoG has a strong response along an edge which may create ambiguous keypoints

Further Outlier Rejection

- The DoG has a strong response along an edge which may create ambiguous keypoints
- If we assume that the DoG is a surface then we can compute the **principle curvatures** (PC)

Further Outlier Rejection

- The DoG has a strong response along an edge which may create ambiguous keypoints
- If we assume that the DoG is a surface then we can compute the **principle curvatures** (PC)
- Along an edge the PC is very low while across an edge the PC is very high

Further Outlier Rejection

- This approach is similar to the Harris detector

Further Outlier Rejection

- This approach is similar to the Harris detector
- First, we compute the Hessian of D

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

Further Outlier Rejection

- This approach is similar to the Harris detector
- First, we compute the Hessian of D

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

- Next, we compute the trace and determinant of H

$$\text{tr}(H) = D_{xx} + D_{yy} = \lambda_1 + \lambda_2$$
$$\det(H) = D_{xx}D_{yy} - (D_{xx})^2 = \lambda_1\lambda_2$$

Further Outlier Rejection

- We then remove outliers by evaluating

$$\frac{\text{tr}(H)^2}{\det(H)} = \frac{(r+1)^2}{r}$$

where $r = \frac{\lambda_1}{\lambda_2}$

Further Outlier Rejection

- We then remove outliers by evaluating

$$\frac{\text{tr}(H)^2}{\det(H)} = \frac{(r+1)^2}{r}$$

  where $r = \frac{\lambda_1}{\lambda_2}$

- This equation is minimum when $r = 1$ and increases with r

Further Outlier Rejection

- We then remove outliers by evaluating

$$\frac{\text{tr}(H)^2}{\det(H)} = \frac{(r+1)^2}{r}$$

  where $r = \frac{\lambda_1}{\lambda_2}$

- This equation is minimum when $r = 1$ and increases with r
- We apply a threshold on r and eliminate keypoints if $r > 10$

Further Outlier Rejection

(a) 729 keypoints

(b) 536 keypoints

- The number of keypoints is further reduced from 729 (a) to 536 (b) using the threshold $r > 10$

Assigning Orientation to Keypoints

- We want to assign a consistent orientation based on local image properties in order to achieve rotation invariance

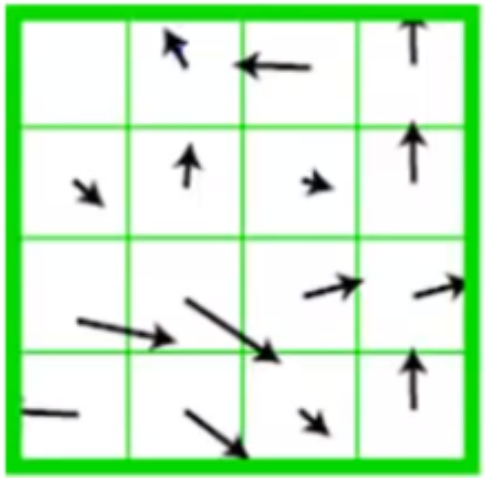
Assigning Orientation to Keypoints

- We want to assign a consistent orientation based on local image properties in order to achieve rotation invariance
- For each smoothed image L, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, is precomputed using pixel differences

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

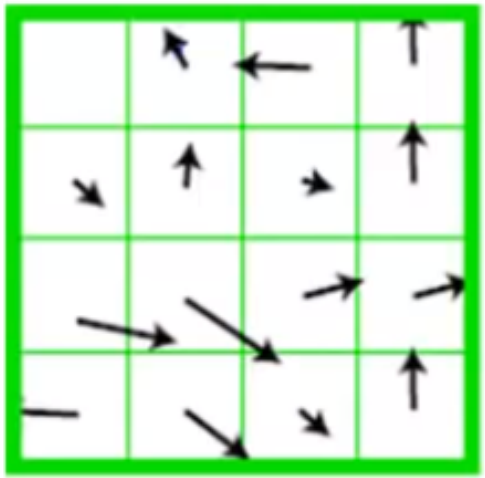
$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

Assigning Orientation to Keypoints



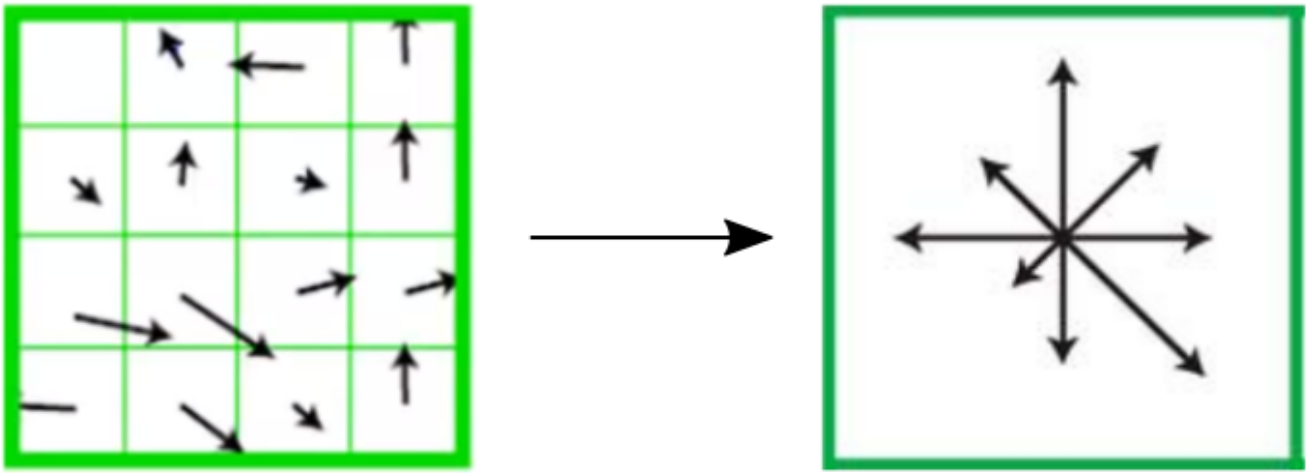
- Then, we create a weighted direction histogram of 36 bins $(10°, 20°, \ldots, 360°)$ in the neighborhood around the keypoint

Assigning Orientation to Keypoints



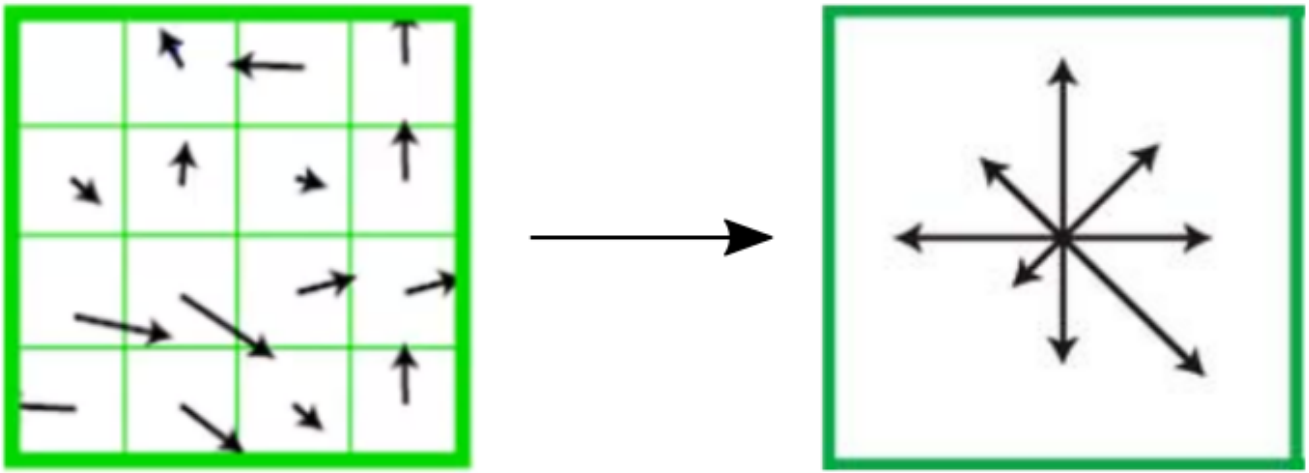
- Then, we create a weighted direction histogram of 36 bins $(10°, 20°, \ldots, 360°)$ in the neighborhood around the keypoint
- The weights are assigned based on the gradient magnitudes

Assigning Orientation to Keypoints



- We select the highest peak as the direction of the keypoint

Assigning Orientation to Keypoints



- We select the highest peak as the direction of the keypoint
- Additional keypoints may be introduced (with different directions) if their peak is within 80% of the highest peak

Local Image Descriptors at Keypoints

- After computing the keypoints we want to create an invariant descriptor

Local Image Descriptors at Keypoints

- After computing the keypoints we want to create an invariant descriptor
- We could do this by extracting a small neighborhood of the intensity values around the keypoint, but this approach is sensitive to noise and lighting conditions

Local Image Descriptors at Keypoints

- After computing the keypoints we want to create an invariant descriptor
- We could do this by extracting a small neighborhood of the intensity values around the keypoint, but this approach is sensitive to noise and lighting conditions
- Instead, the gradient orientation histograms are used which are much more stable (e.g. a small change in illumination results in a negligible change in gradient)

Motivation: Similarity to the IT Cortex

- Complex neurons respond to a gradient at a particular orientation

Motivation: Similarity to the IT Cortex

- Complex neurons respond to a gradient at a particular orientation
- The location of a feature can shift over a small receptive field

Motivation: Similarity to the IT Cortex

- Complex neurons respond to a gradient at a particular orientation
- The location of a feature can shift over a small receptive field
- The function of the cells allow for matching and recognition of 3D objects from a wide range of viewpoints (Edelman, Intrator, and Poggio, 1997)

Motivation: Similarity to the IT Cortex

- Complex neurons respond to a gradient at a particular orientation
- The location of a feature can shift over a small receptive field
- The function of the cells allow for matching and recognition of 3D objects from a wide range of viewpoints (Edelman, Intrator, and Poggio, 1997)
- Experimental results show better recognition accuracy for 3D objects rotated by up to 20°

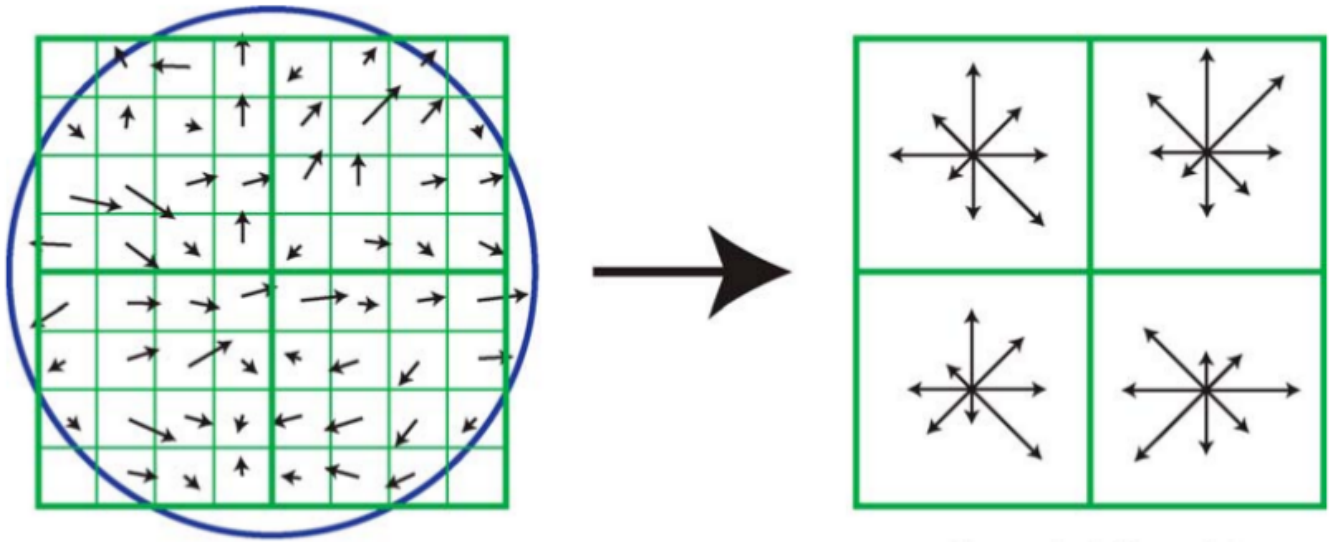
Extraction of Local Image Descriptors at Keypoints

- For each keypoint we have the dominate orientation computed in the orientation assignment stage

Extraction of Local Image Descriptors at Keypoints

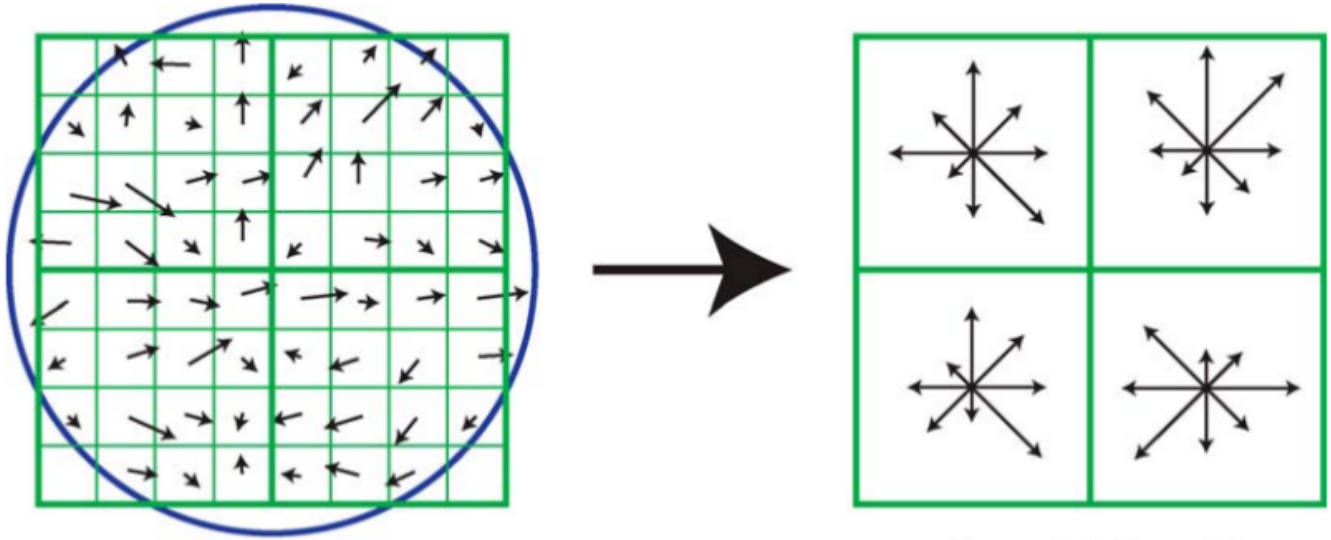
- For each keypoint we have the dominate orientation computed in the orientation assignment stage
- We compute the relative orientation and magnitude w.r.t the dominate orientation of the keypoint in a 16×16 neighborhood

Extraction of Local Image Descriptors at Keypoints



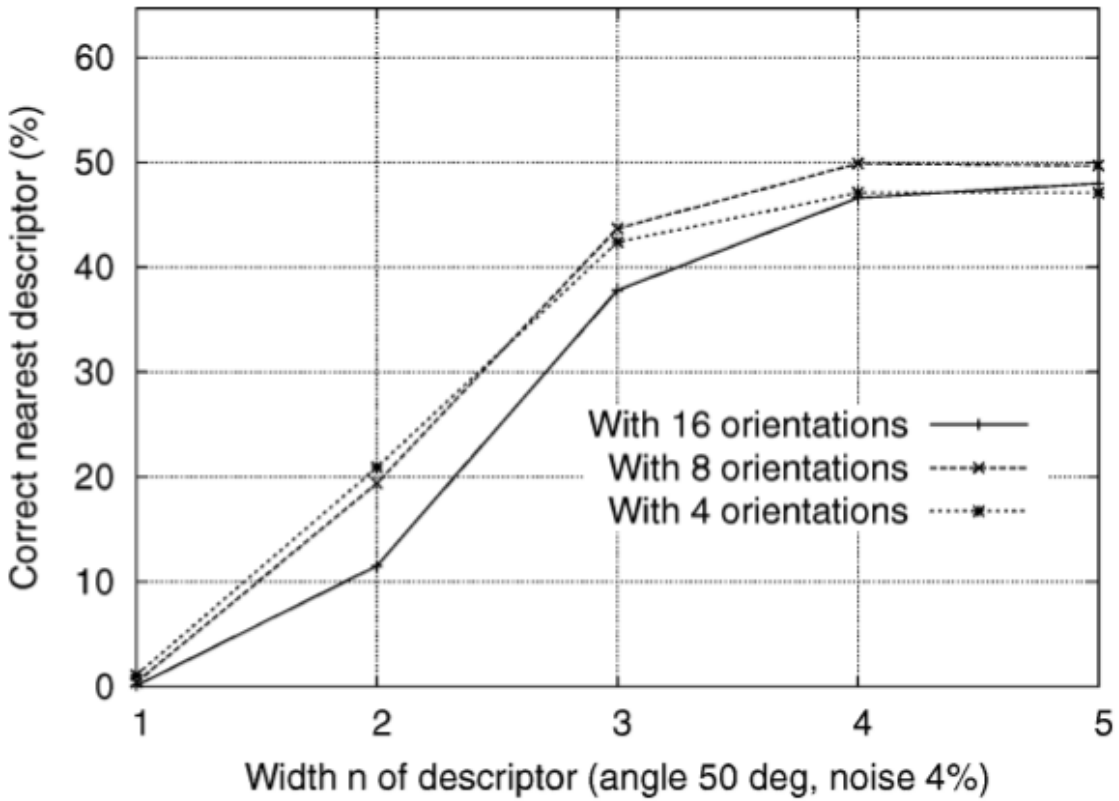
- Next, we form a weighted histogram (8 bins) for 4×4 regions

Extraction of Local Image Descriptors at Keypoints



- Next, we form a weighted histogram (8 bins) for 4×4 regions
- Then, the 16 histograms are concatenated into one long vector of 128 dimensions

Width of the Descriptor Region



Descriptor Post-Processing

- We normalize the 128-dimensional vector to make it a unit vector

Descriptor Post-Processing

- We normalize the 128-dimensional vector to make it a unit vector
- For nonlinear intensity transforms we can bound the unit vector to a maximum of 0.2 (i.e. remove large gradients) and renormalize

Keypoint Matching

- We match the keypoints against a database obtained from the training images

Keypoint Matching

- We match the keypoints against a database obtained from the training images
- To do this we find the nearest neighbors, i.e. a keypoint with minimum Euclidean distance

Keypoint Matching

- We match the keypoints against a database obtained from the training images
- To do this we find the nearest neighbors, i.e. a keypoint with minimum Euclidean distance
- An efficient nearest neighbors match considers the ratio of the distance between the first and second best match using a threshold of 0.8

Matching Local Features

Matching Local Features

Image 1

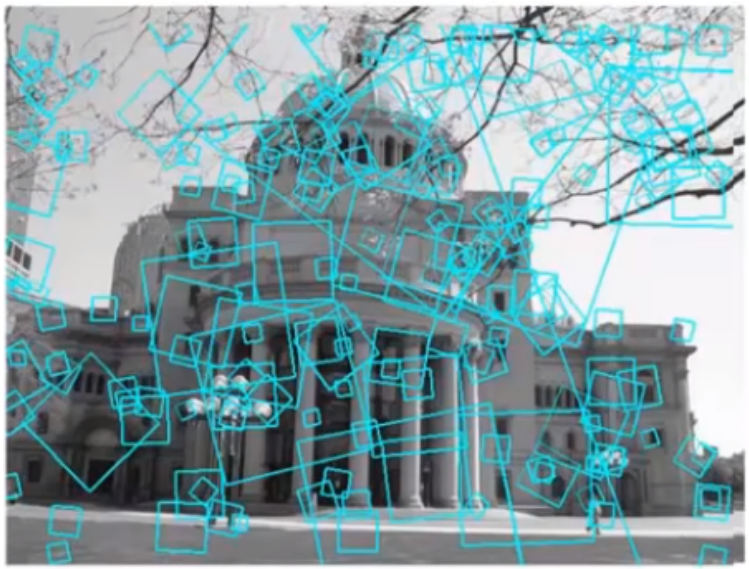


Image 2

Matching Local Features

Image 1

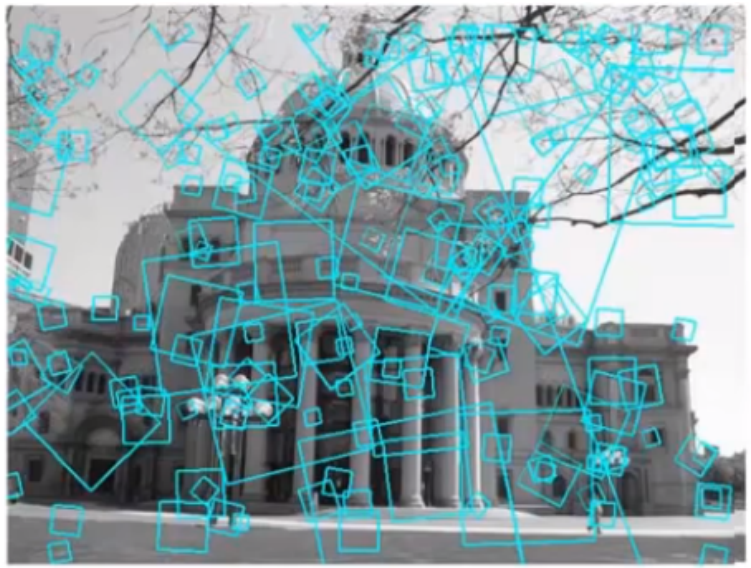
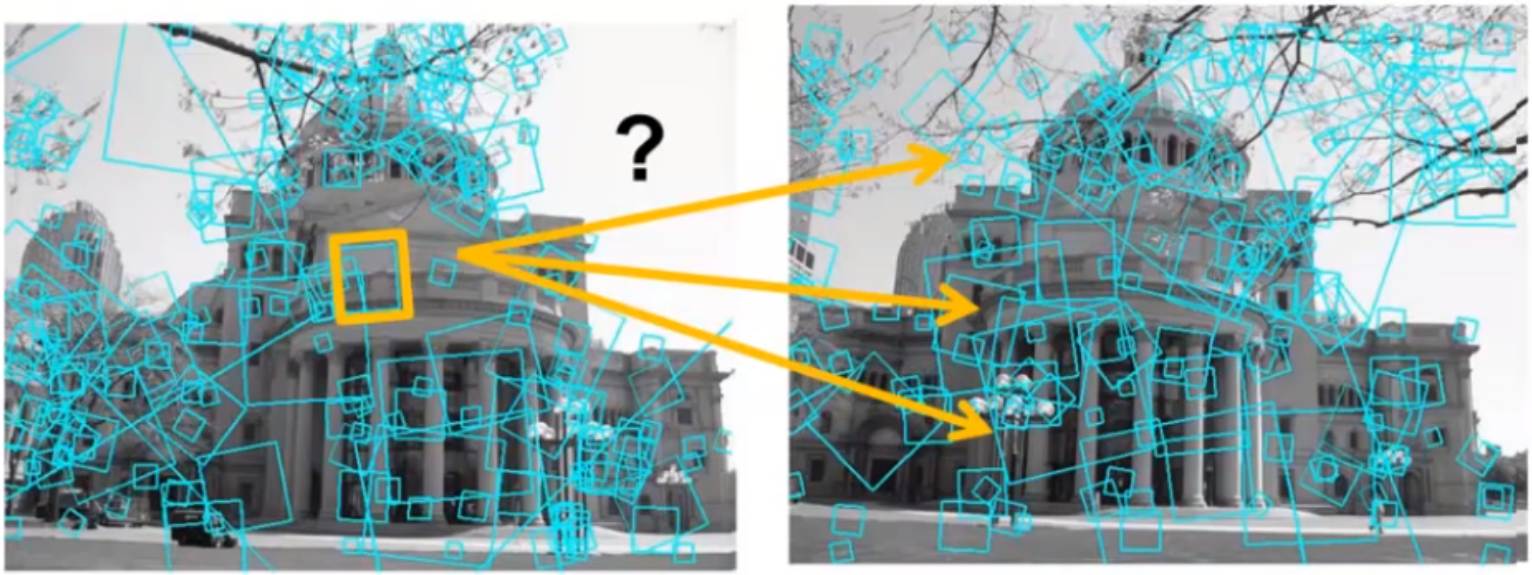


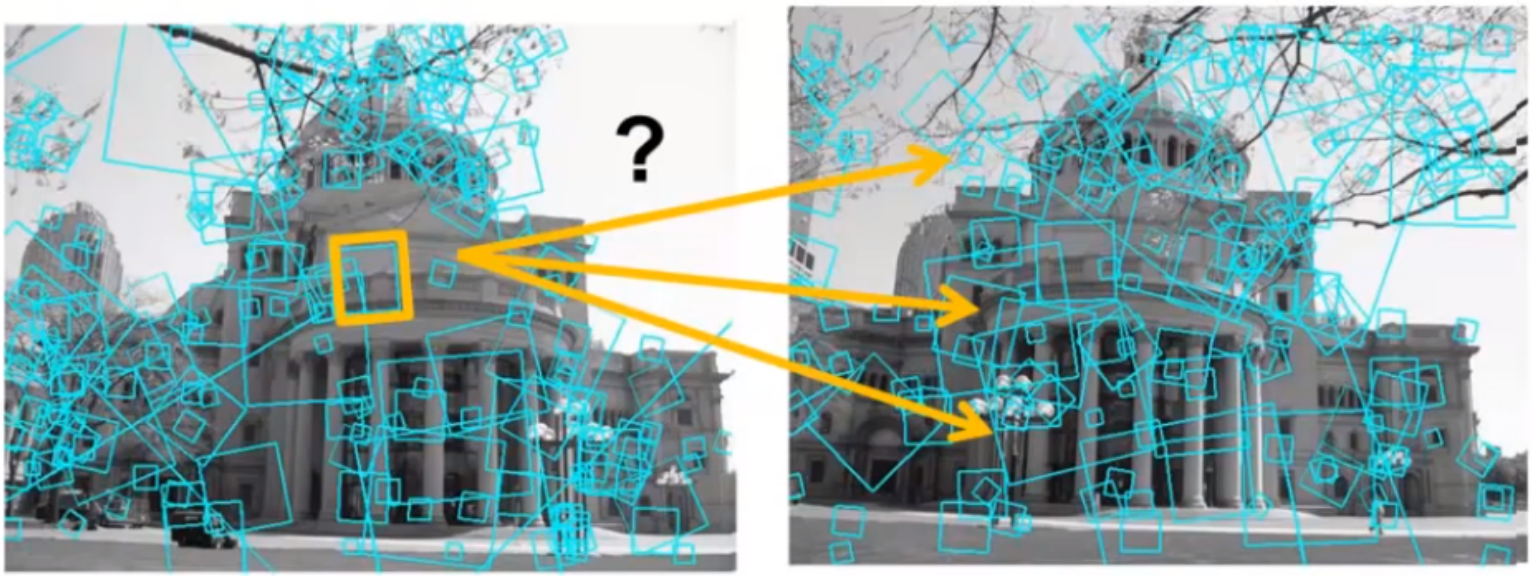
Image 2

Matching Local Features



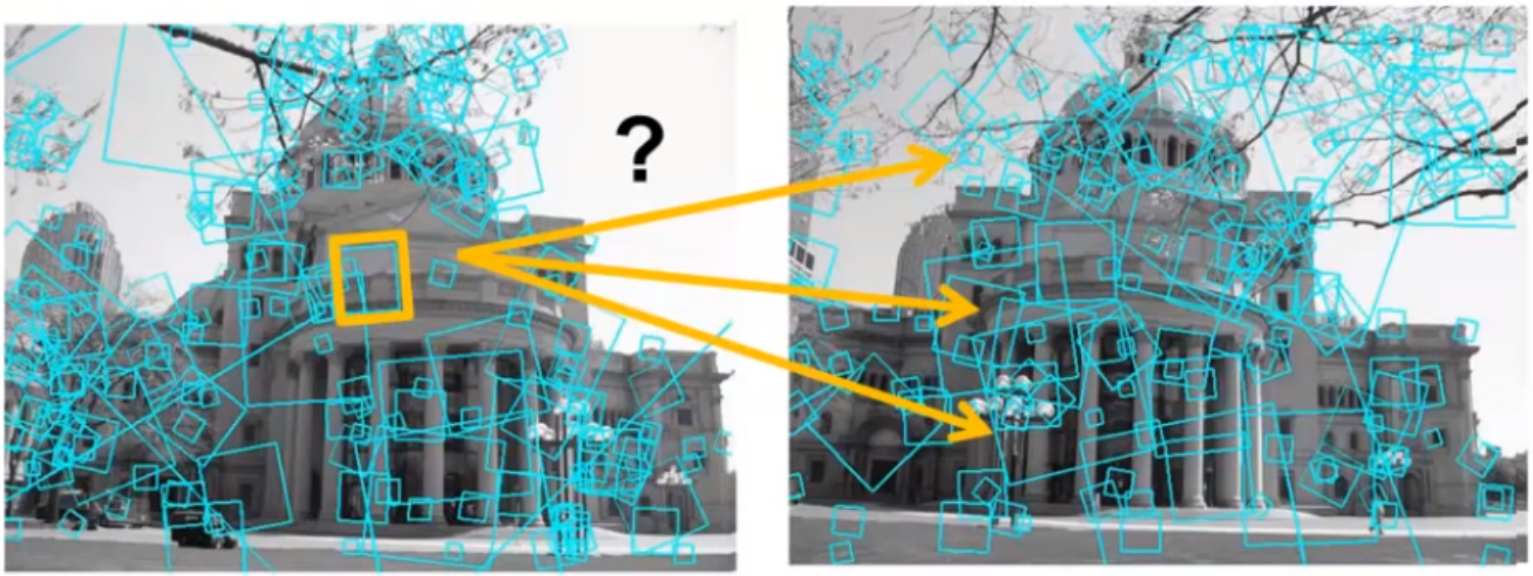
- A simple approach to finding candidate matches is to compare them all and take the closest (or k closest within a threshold distance)

Matching Local Features



- A simple approach to finding candidate matches is to compare them all and take the closest (or k closest within a threshold distance)
- However, at what distance do we have a good match?

Matching Local Features



- To add robustness to the matching process, we consider the ratio of the distance to the first best match and the second best match

Matching Local Features

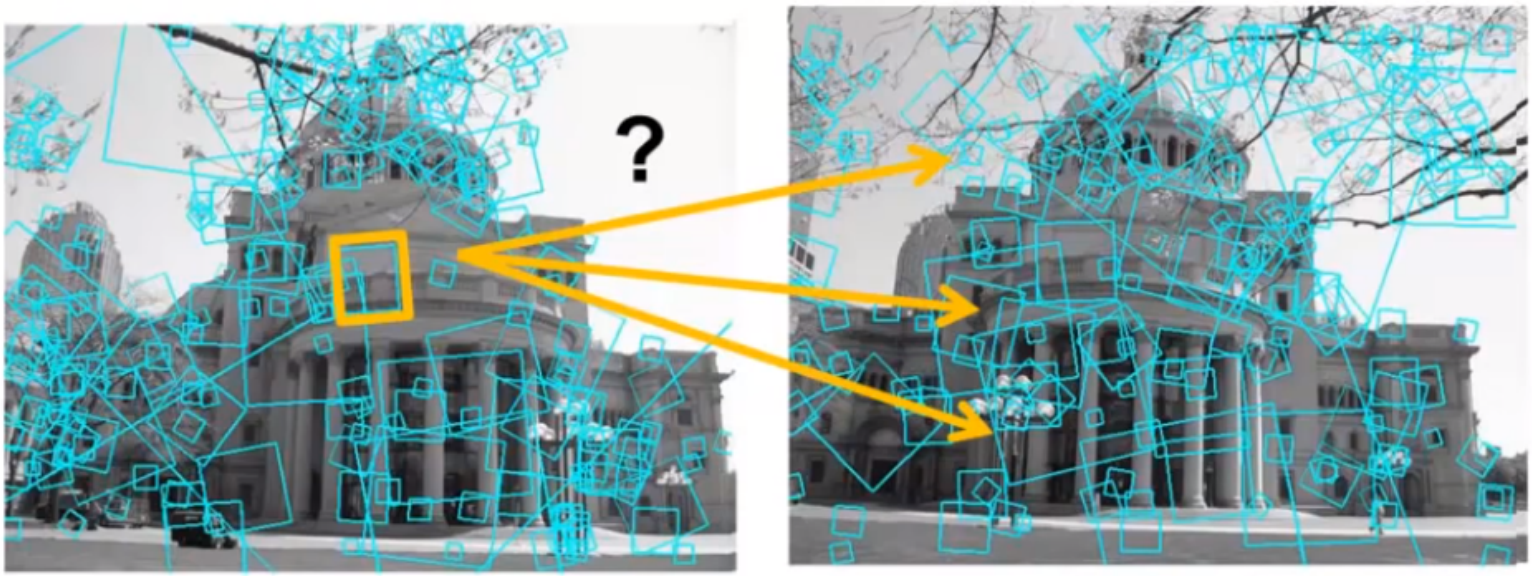
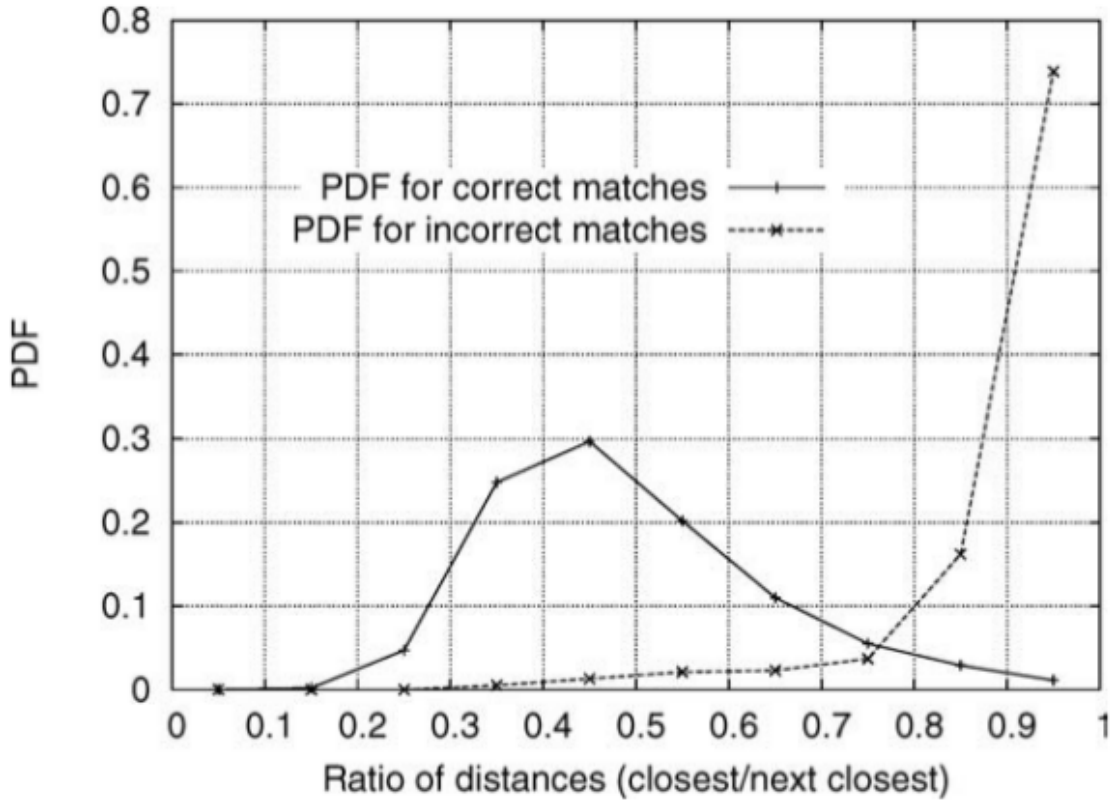


Image 1 Image 2

- To add robustness to the matching process, we consider the ratio of the distance to the first best match and the second best match
- If the ratio is greater than 0.8 then we take the match, otherwise we do not take the match

The Ratio of Distances Between Nearest Descriptors



Summary

- SIFT consists of both a feature detector and descriptor

Summary

- SIFT consists of both a feature detector and descriptor
- The main idea of the detector is to find extrema (minima or maxima) across a spectrum of scales

Summary

- SIFT consists of both a feature detector and descriptor
- The main idea of the detector is to find extrema (minima or maxima) across a spectrum of scales
- The robustness of the descriptor makes it suitable for performing real-time object recognition in cluttered scenes

References

- D.G. Lowe, "Distinctive Image Features From Scale-Invariant Keypoints." *International Journal of Computer Vision*, 60.2, 2004, pp. 91-110.