

Computer Vision and Machine Learning (Keypoint detector and descriptor)

Bhabatosh Chanda
chanda@isical.ac.in

4/25/2023

Computer Vision -- Intro

1

Binocular stereo reconstruction

1. Compute image features.
2. Compute feature descriptors.
3. Find initial matches.
4. Compute fundamental matrix.
5. Refine matches.
6. Estimate essential matrix.
7. Decompose essential matrix.
8. Estimate 3D points.

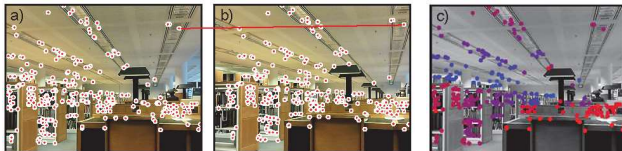
4/25/2023

Computer Vision

2

Correspondence problem

- How to detect points in the scene (object) whose coordinates need to be determined.
- How to establish correspondence between points (in different camera frames) which are images of same scene point.
- How to perform reliable and efficient search.



4/25/2023

Computer Vision

3

Detector and descriptor

- Harris corner detector
- Histogram of Oriented Gradients (HOG)
- Scale Invariant Feature Transform (SIFT)

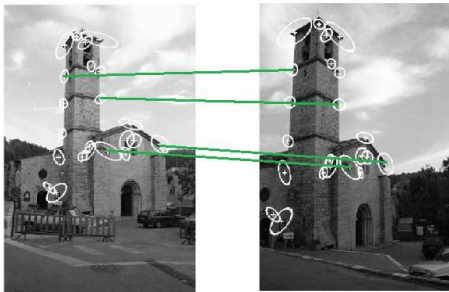
4/25/2023

Computer Vision

4

Motivation: Matching Problem

Vision tasks such as stereo and motion estimation require finding corresponding features across two or more views.



Robert Collins, Penn State Univ.

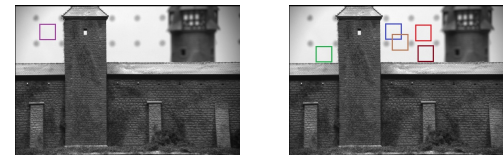
4/25/2023

Computer Vision

5

Not all Patches Created are Equal!

- A blindly picked up patch may not help perform the task



- Feature-less or attribute-less patches are highly probable in image and match with many such similar ones



4/25/2023

Computer Vision

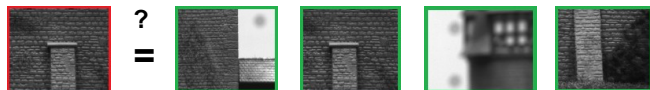
6

Motivation: Patch Matching

- Elements to be matched are image patches of fixed size



- Task: find the best (most similar) patch in a second image



4/25/2023

Computer Vision

7

Motivation: Patch Matching

- Elements to be matched are image patches of fixed size



- A distinctive patch is a good patch for matching

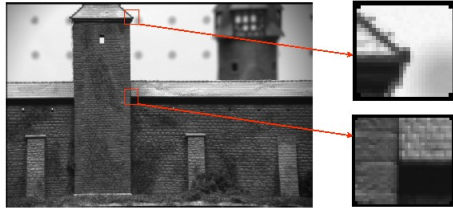


4/25/2023

Computer Vision

8

What is corner?



- High curvature point on contour (edge)
- Junction of contours
- Usually stable features with respect to view points
- Large variation in neighborhood of the point in almost all directions

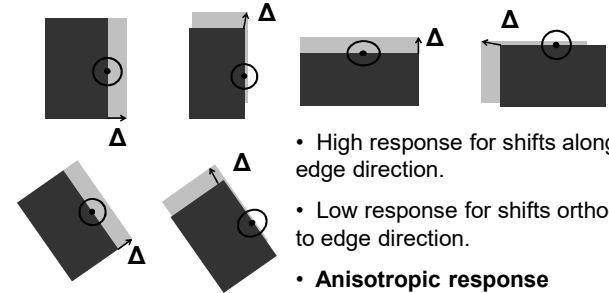
4/25/2023

Computer Vision

9

Harris corner detector

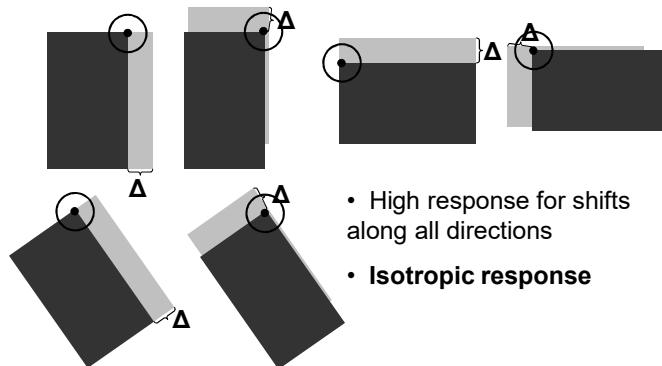
- Key idea: Measure changes over a neighborhood due to a shift and then analyze dependency on shift direction.
- Direction dependency of the response for lines



- High response for shifts along the edge direction.
- Low response for shifts orthogonal to edge direction.
- **Anisotropic response**

Key idea: *continued ...*

- Orientation dependence of the shift response for corners



- High response for shifts along all directions
- **Isotropic response**

Harris corner: formulation

- An image patch or neighborhood W is shifted by a shift vector $\Delta = [\Delta x, \Delta y]$.
- A corner does not have the **aperture problem** and therefore should show high shift response for all orientation of Δ .
- Sum of squared intensity difference between the original and the shifted image over the neighborhood W is

$$S_W(\Delta) = \sum_{(x_i, y_i) \in W} (f(x_i, y_i) - f(x_i + \Delta x, y_i + \Delta y))^2$$

Harris corner: formulation

- Sum of squared intensity difference between original and shifted image over W is

$$S_W(\Delta) = \sum_{(x_i, y_i) \in W} (f(x_i, y_i) - f(x_i + \Delta x, y_i + \Delta y))^2$$

- Apply Taylor expansion

$$f(x_i + \Delta x, y_i + \Delta y) = f(x_i, y_i) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial^2 f}{\partial x^2} \frac{(\Delta x)^2}{2!} + \frac{\partial^2 f}{\partial y^2} \frac{(\Delta y)^2}{2!} + \dots$$

$$f(x_i + \Delta x, y_i + \Delta y) \approx f(x_i, y_i) + \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

Harris corner [Continued ...]

$$\begin{aligned} S(x, y, \Delta) &= \sum_{(x_i, y_i) \in W} \left(f(x_i, y_i) - f(x_i, y_i) - \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\ &= \sum_{(x_i, y_i) \in W} \left(- \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\ &= \sum_{(x_i, y_i) \in W} \left(\begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\ &= \sum_{(x_i, y_i) \in W} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} \\ \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned}$$

Harris corner [Continued ...]

$$\begin{aligned} &= \sum_{(x_i, y_i) \in W} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} \\ \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \left(\sum_{(x_i, y_i) \in W} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} \\ \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \sum_{(x_i, y_i) \in W} \left(\frac{\partial f(x_i, y_i)}{\partial x} \right)^2 & \sum_{(x_i, y_i) \in W} \frac{\partial f(x_i, y_i)}{\partial x} \frac{\partial f(x_i, y_i)}{\partial y} \\ \sum_{(x_i, y_i) \in W} \frac{\partial f(x_i, y_i)}{\partial x} \frac{\partial f(x_i, y_i)}{\partial y} & \sum_{(x_i, y_i) \in W} \left(\frac{\partial f(x_i, y_i)}{\partial y} \right)^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \Delta^T \mathbf{A}_W(x, y) \Delta \end{aligned}$$

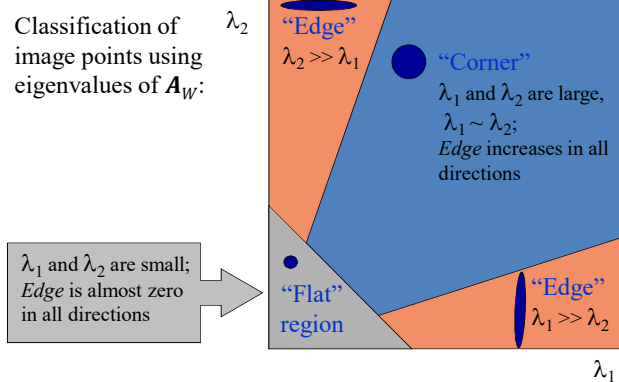
The matrix \mathbf{A}_W is called the **Harris matrix**.

Harris matrix

- The Harris matrix \mathbf{A}_W is symmetric and positive semi-definite.
- PCA of \mathbf{A}_W gives eigen vector (e_1, e_2) and eigen value (λ_1, λ_2).
- Three distinct situations:
 - Both λ_1 and λ_2 are small \Rightarrow a flat region
 - One λ is large and other is small \Rightarrow existence of edge
 - Both λ_1 and λ_2 are large \Rightarrow existence of corner

Harris Detector: Implementation

Classification of image points using eigenvalues of \mathbf{A}_W :



Harris Detector: Implementation

Measure of corner response:

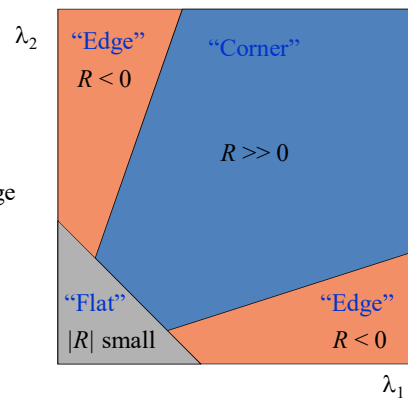
$$R = \det \mathbf{A}_W - k(\text{trace} \mathbf{A}_W)^2$$

where $\det \mathbf{A}_W = \lambda_1 \lambda_2$
 $\text{trace} \mathbf{A}_W = \lambda_1 + \lambda_2$

(k – empirical constant, $k = 0.04 - 0.06$)

Harris Detector: Mathematics

- R depends only on eigenvalues of \mathbf{A}_W
- R is large for a **corner**
- R is negative with large magnitude for an **edge**
- $|R|$ is small for a **flat** region

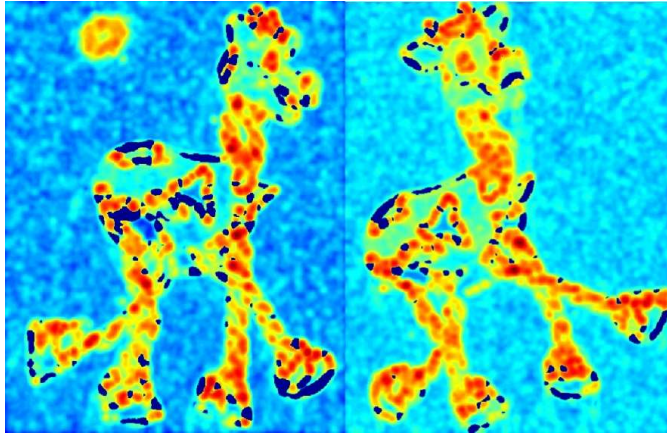


Harris Detector: Workflow



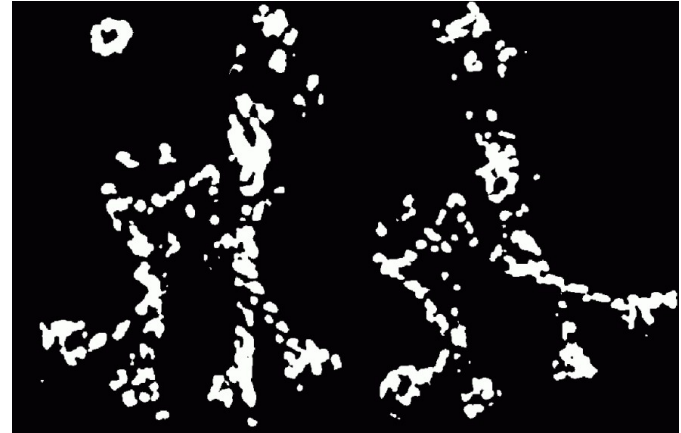
Harris Detector: Workflow

Compute corner response R



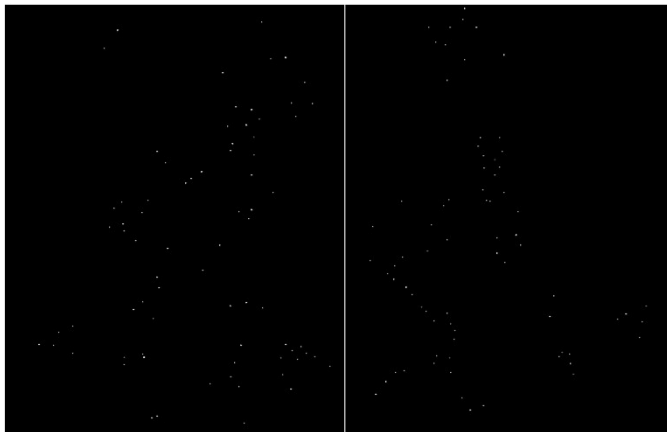
Harris Detector: Workflow

Find points with large corner response: $R > \text{threshold}$



Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow



Histogram of Oriented Gradient (HOG)

HOG feature extraction

- Compute horizontal gradient $\frac{\partial f}{\partial x}$ and vertical gradient $\frac{\partial f}{\partial y}$ after smoothing
- Compute gradient orientation $\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$ and magnitude $|\nabla f| = \sqrt{\frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}}$
 - For color image, pick the color channel with the highest gradient magnitude for each pixel.

4/25/2023

Computer Vision

25

HOG feature: Example

For a 64x128 image,

- Divide the image into 16x16 blocks of 50% overlap.
 - 7x15=105 blocks in total
- Each block should consist of 2x2 cells with size 8x8.
- Quantize the gradient orientation into 9 bins
 - The vote is the gradient magnitude
 - Interpolate votes between neighbouring bin centre.
 - The vote can also be weighted with Gaussian kernel to rationalize the pixels near the edges of block.
- Concatenate histograms (dimension: 105x4x9=3,780)

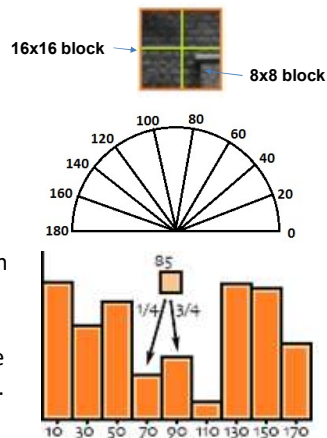
4/25/2023

Computer Vision

26

Votes

- Each block consists of 2x2 cells with size 8x8
- Quantize the gradient orientation into 9 bins (0-180)
- The vote is the gradient magnitude
- Interpolate votes linearly between neighbouring bin centres.
- The vote can also be weighted with Gaussian to down weight the pixels near the edges of the block.

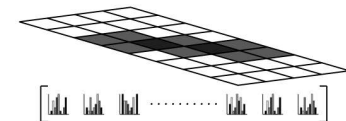


4/25/2023

Computer Vision

Final Feature Vector

- Concatenate histograms
 - Make it a 1D vector of length 3780.



- Visualization



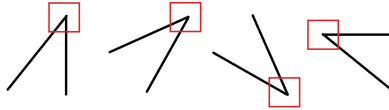
4/25/2023

Computer Vision

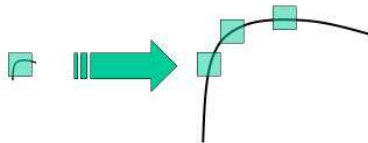
28

Harris corner detector

- Harris corner detector is rotation invariant



- But it is not scale invariant (as corners are not!!)



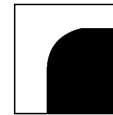
- Invariance:** features (*transform* (image)) = features (image)

4/25/2023

Computer Vision

29

Corner at different scales

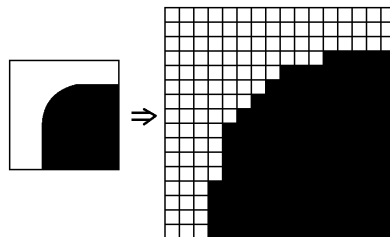


4/25/2023

Computer Vision

30

Corner at different scales

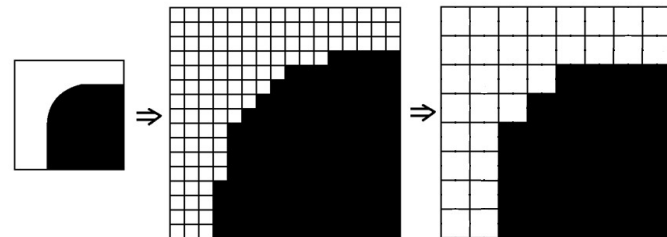


4/25/2023

Computer Vision

31

Corner at different scales

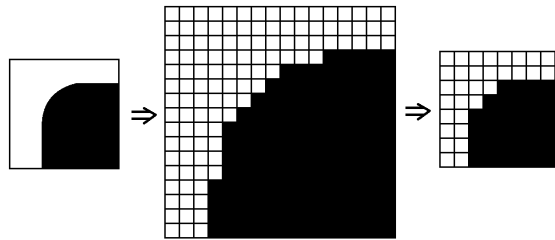


4/25/2023

Computer Vision

32

Corner at different scales

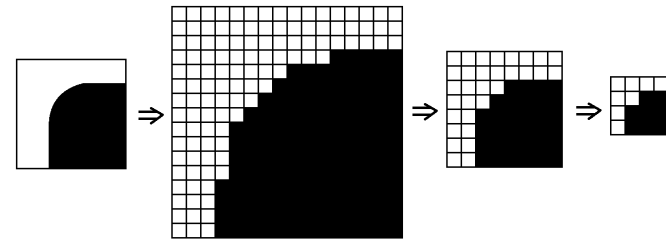


4/25/2023

Computer Vision

33

Corner at different scales

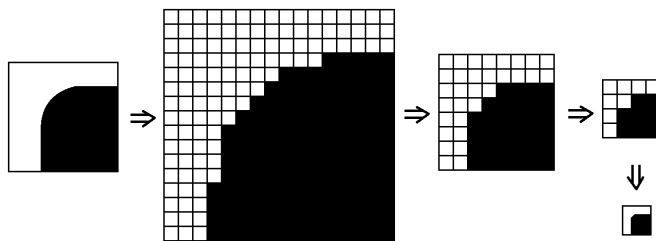


4/25/2023

Computer Vision

34

Corner at different scales



4/25/2023

Computer Vision

35

Type of invariance

- Illumination



4/25/2023

Computer Vision

36



Type of invariance

- Illumination
- Scale



4/25/2023

Computer Vision

37

Type of invariance

- Illumination
- Scale
- Rotation



4/25/2023

Computer Vision

38

Type of invariance

- Illumination
- Scale
- Rotation
- Affine



4/25/2023

Computer Vision

39

SIFT

- Scale Invariant Feature Transform (SIFT)
- David Lowe, **Distinctive Image Features from Scale-Invariant Keypoints**, IJCV, 2004.
- Lowe aimed to create a **descriptor** that was robust to the variations corresponding to typical viewing conditions. **Descriptor is the most-used part of SIFT.**
- SIFT transforms image data into scale-invariant coordinates (location) corresponding to local features.

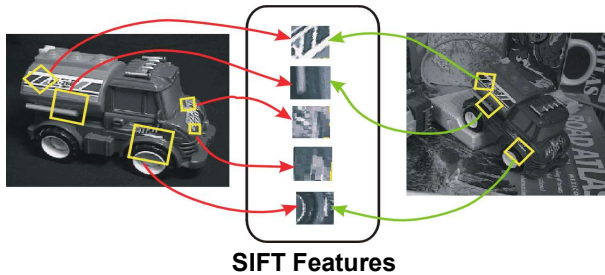
4/25/2023

Computer Vision

40

Idea of SIFT

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



4/25/2023

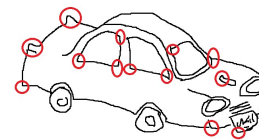
Computer Vision

41

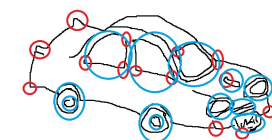
Corner vs. SIFT



Car: original



Corner only



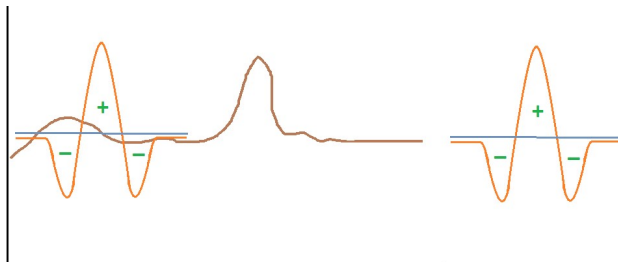
SIFT

4/25/2023

Computer Vision

42

Correlation with normalized LoG

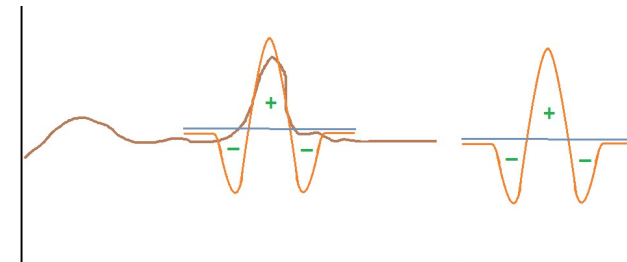


4/25/2023

Computer Vision

43

Correlation with normalized LoG

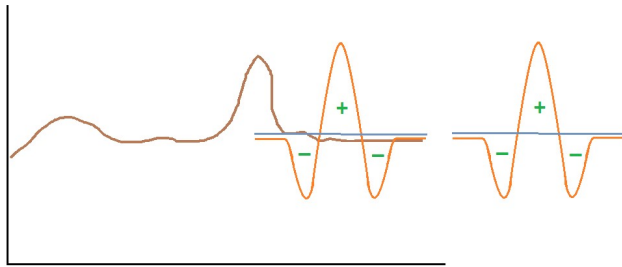


4/25/2023

Computer Vision

44

Correlation with normalized LoG

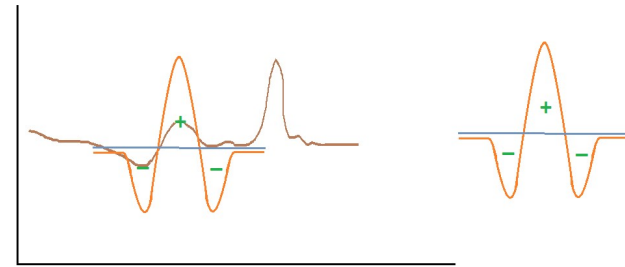


4/25/2023

Computer Vision

45

Correlation with normalized LoG



4/25/2023

Computer Vision

46

Correlation and convolution

- Correlation between $f(x)$ and $g(x)$ is defined as

$$h(x) = \sum_{u \in D_g} f(u)g(u + x)$$

- Convolution between $f(x)$ and $g(x)$ is defined as

$$h(x) = \sum_{u \in D_g} f(u)g(u - x)$$

- If $h(x)$ be symmetric about y-axis, correlation may obtained by computing convolution.

4/25/2023

Computer Vision

47

SIFT algorithm: Major steps

1. Scale-space extrema detection

- Search over multiple scales and image locations.

2. Keypoint localization

- Fit a model to determine location and scale. Select keypoints based on a measure of stability.

3. Orientation assignment

- Compute best orientation(s) for each keypoint region.

4. Keypoint description

- Use local image gradients at selected scale and rotation to describe each keypoint region.

4/25/2023

Computer Vision

48

Step 1. Scale-space extrema detection

- **Goal:** Identify locations and scales that can be repeatedly found under different views of the same scene or object.
- **Method:** search for stable features across multiple scales using a continuous function of scale.
 - Prior work has shown that under a variety of assumptions, the best function is a **Gaussian function**.
 - The scale space of an image is a function $L(x, y, k\sigma)$ that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.

4/25/2023

Computer Vision

49

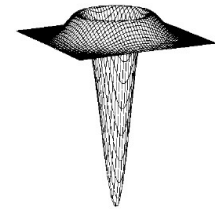
Method: Scale-space extrema detection

- Find the points, whose surrounding patches (at some scale) are distinctive.
- A plausible method is to convolve the image $I(x, y)$ with Laplacian of Gaussian.
 - Scale normalized (x by scale²)
 - Proposed by Lindeberg (1994)

$$\nabla^2 S = \nabla^2 (G_\sigma * I) = \nabla^2 G_\sigma * I$$

$$\text{where } \nabla^2 G(x, y, \sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

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



4/25/2023

Computer Vision

50

Method: Scale-space extrema detection

- Gaussian is an *ad hoc* solution of heat diffusion equation

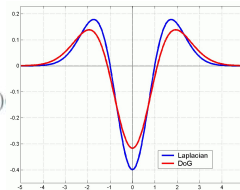
$$\frac{\partial^2 G}{\partial x^2} = \sigma \nabla^2 G$$

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

- An approximation to the scale-normalized Laplacian of Gaussian is DoG:

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

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

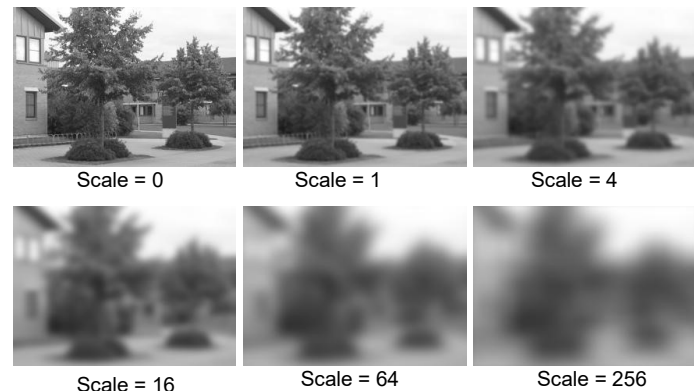


4/25/2023

Computer Vision

51

Scale-space representation

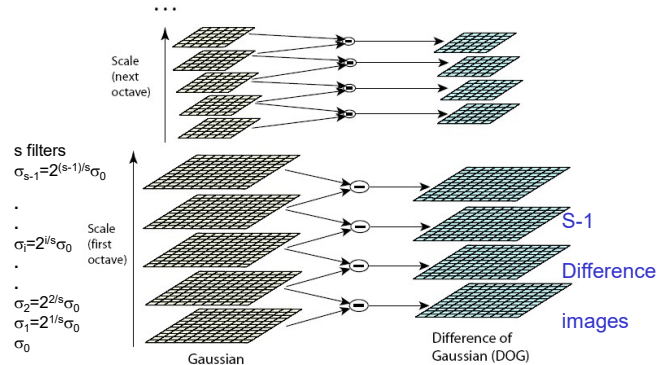


4/25/2023

Computer Vision

52

Lowe's pyramid scheme



The parameter s determines the number of images per octave.

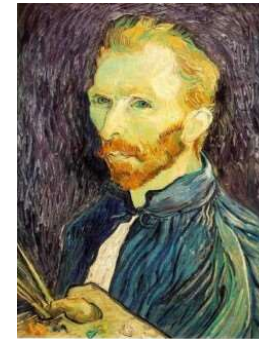
number of octaves = 4, number of scale levels = 5, $\sigma_0 = 1.6$, $k = \sqrt{2}$

4/25/2023

Computer Vision

53

Example: Subsampling with Gaussian smoothing



Original



G 1/2



G 1/4

4/25/2023

54

Lowe's Pyramid Scheme

- Scale space is separated into **octaves**:
 - Octave 1 uses scale σ
 - Octave 2 uses scale 2σ
 - so on
- In each octave, the initial image is repeatedly convolved with Gaussians to produce a set of scale space images.
- Adjacent Gaussians are subtracted to produce the DOG
- After each octave, the Gaussian image is down-sampled by a factor of 2 to start the next level.

4/25/2023

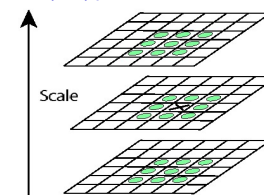
Computer Vision

55

Step 2: Keypoint localization

- Detect maxima and minima of difference-of-Gaussian in scale space
- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below

(s-1) difference images.
top and bottom ignored.
(s-3) planes searched.



For each max or min found, output is the **location** and the **scale**.

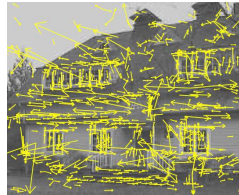
4/25/2023

56

Keypoint localization



(a) 233x189 image



(b) 832 DOG extrema

- Too many keypoints, some are unstable:
 - points with low contrast (sensitive to noise)
 - points that are localized along an edge

4/25/2023

Computer Vision

57

Keypoint localization

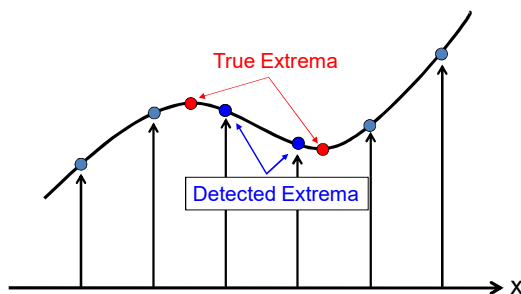
- Once a keypoint candidate is found, perform a detailed fit to nearby data to determine
 - Exact location, scale, and ratio of principal curvatures
- In initial work, keypoints are found at location and scale of a central sample point.
- In refinement work, they fit a 3D quadratic function to improve interpolation accuracy.
- The Hessian matrix was used to eliminate edge responses.

4/25/2023

58

Keypoint localization

■ The Problem:



4/25/2023

Computer Vision

59

Keypoint Localization

Low contrast points elimination:

- Fit keypoint at x to nearby data using quadratic approximation

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

- $x = (x, y, \sigma)$ and $h = (\Delta x, \Delta y, \Delta \sigma)$
- Calculate the local extrema $x + h = \hat{x}$ of the fitted function.

$$\hat{x} = - \left[\frac{\partial^2 D}{\partial x^2} \right]^{-1} \frac{\partial D}{\partial x}$$

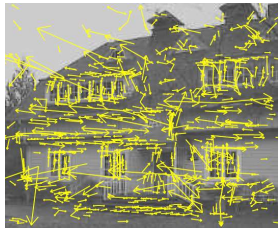
- Discard local extrema $|D(\hat{x})| < 0.03$

4/25/2023

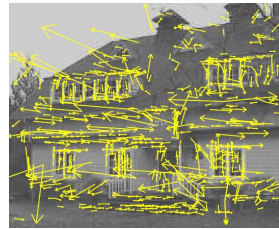
Computer Vision

60

Keypoint Localization



(a) 832 DOG extrema



(b) 729 after deleting weak extrema

729 out of 832 are left after contrast thresholding

Eliminating the Edge Response

- Reject flats: $|D(\hat{\mathbf{x}})| < 0.03$
- Reject edges: $\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$
- Let α and β be the eigenvalues with $\alpha > \beta$.

$$\text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

- Let $r = \alpha/\beta$, so $\alpha = r\beta$ and $r > 1.0$

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r},$$

$(r+1)^2/r$ is at a min when the 2 eigenvalues are equal.

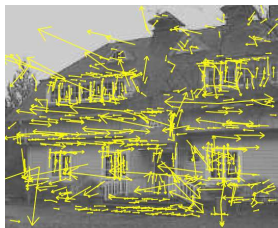
- Good choice $r > 10$

4/25/2023

Computer Vision

62

Keypoint Localization



(a) 729 DOG extrema



(b) 536 after ratio thresholding

536 out of 832 are left after contrast and ratio thresholding

Keypoint Localization



(a)



(b)

832 keypoints



(c)

729 keypoints



(d)

536 keypoints

4/25/2023

Computer Vision

64

Step 3: Orientation assignment

- Assign an orientation to each keypoint, the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.
- A neighbourhood is taken around the keypoint location depending on the scale.
- Compute magnitude and orientation of gradient on the Gaussian smoothed images.

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

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

4/25/2023

Computer Vision

65

Orientation assignment

- A histogram is formed by quantizing the (360 degrees) orientations into 36 bins.
- Vote is gradient magnitude and Gaussian-weighted window with $\sigma = 1.5$ times the scale of the keypoint.
- Peaks (and also 80% of it) in the histogram are considered to compute orientation of the patch.
- At the same location, there could be multiple keypoints with different orientations.

4/25/2023

Computer Vision

66

Step 4: Keypoint Descriptors

- At this point, each keypoint has
 - location
 - scale
 - orientation
- Next is to compute a descriptor for the local image region about each keypoint that is
 - highly distinctive
 - invariant (as much as possible) to variations or changes in viewpoint and illumination

4/25/2023

67

Normalization

- Rotate the window to standard orientation
- Scale the window size based on the scale at which the point was found.

Remaining goal:

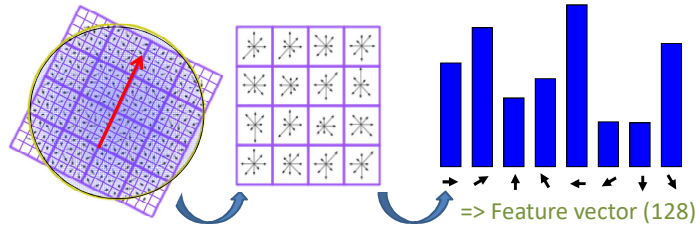
- Define **local** descriptor invariant to remaining variations:
 - Illumination
 - 3D Viewpoint

4/25/2023

68

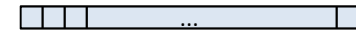
Keypoint descriptor

- Create 16 gradient histograms (8 bins)
 - Weighted by magnitude and Gaussian window (σ is half the window size)
 - Histogram and gradient values are interpolated and smoothed



Lowe's Keypoint Descriptor

- Use the **normalized** region about the keypoint
- Compute gradient magnitude and orientation at each point in the region
- **Weight them by a Gaussian** window overlaid on the circle
- Create an **orientation histogram** over the 4X4 subregions of the window
- 4X4 descriptors over 16X16 sample array were used.
4X4 times 8 directions gives a **vector of 128 values**.



4/25/2023

70

Thank you !
Any question?

4/25/2023

Computer Vision – Intro

71