Chapter 04
Interest Point Detection and Matching

Lin ZHANG
School of Computer Science and Technology
Tongji University

Content

- General requirements for interest point detectors and descriptors
- Harris feature
  - Point detector
  - Feature descriptor
  - The distance between two descriptors
- SIFT feature
  - Point detector
  - Feature descriptor
  - The distance between two descriptors
- ORB feature
  - Point detector
  - Feature descriptor
  - The distance between two descriptors
- Feature point matching
- Practice

General requirements for interest point detectors and descriptors

- For point detectors
  - **Locality**: The positions of feature points need to be easily and accurately locatable. For example, edge points in an image do not exhibit good localization because moving along the edge direction passes through points with highly similar appearances
  - **Sparsity**: The number of feature points in an image should be relatively sparse compared to the total number of pixels. If the detected feature points are too dense, it will significantly increase the computational cost of subsequent processing
  - **Stability to illumination changes**: When the environmental lighting conditions change, we expect the feature point detection algorithm to still identify the same feature points
  - **Stability to geometric transformations**: When the camera shooting perspective changes, the image plane undergoes corresponding geometric transformations, and we hope that the feature point detection algorithm can still detect the corresponding feature points

General requirements for interest point detectors and descriptors

- For point descriptors
  - **High discriminability**: Suppose $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two image feature points (from different images), and $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are their respective feature descriptors. If $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ correspond to the same point in the physical scene, we expect $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ to be identical; if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ correspond to different points in the physical scene, we expect the distance between $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ to be large
  - **Stability to illumination changes**: When the environmental lighting conditions change, we hope that the feature descriptors constructed for the corresponding feature points can remain unchanged
  - **Stability to geometric transformations**: When the camera shooting perspective changes, the image plane undergoes corresponding geometric transformations, and we hope that the feature descriptors constructed for the corresponding feature points can remain unchanged

General requirements for interest point detectors and descriptors

An example

NASA Mars Rover Images

General requirements for interest point detectors and descriptors

An example (Look for tiny colored squares)

NASA Mars Rover images with SIFT matches

Content

- General requirements for interest point detectors and descriptors
- Harris feature
  - Point detector
  - Feature descriptor
  - Distance between two descriptors
- SIFT feature
  - Point detector
  - Feature descriptor
  - Distance between two descriptors
- ORB feature
  - Point detector
  - Feature descriptor
  - Distance between two descriptors
- Feature point matching
- Practice

Harris feature—point detector

My office,

5:30PM, Sep. 18, 2011

Harris feature—point detector

- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive

C. Harris and M. Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988

Harris feature—point detector (basic idea)

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity

"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Harris feature—point detector (basic idea)

Difference = 3

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (basic idea)

Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (basic idea)

Difference = 5

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (basic idea)

Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (basic idea)

Difference = 3

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (basic idea)

Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (basic idea)

Difference = 3

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (basic idea)

Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.

Harris feature—point detector (math)

Consider the image as a 2D function $f(x,y):\mathbb{R}^2 \rightarrow \mathbb{R}$

Change in appearance of a local patch (defined by a window W) centered at $\boldsymbol{x}$ for the shift $(\Delta x, \Delta y)$:

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

Window function

Intensity

Shifted intensity

Window function W=

or

1 in window, 0 outside

Gaussian

Harris feature—point detector (math)

$$s_W(\Delta x,\Delta y)=\sum_{(x_i,y_i)\in w}(f(x_i,y_i)-f(x_i+\Delta x,y_i+\Delta y))^2 \qquad (1)$$

$$\simeq\sum_{(x_i,y_i)\in W}\left(f(x_i,y_i)-f(x_i,y_i)-\left(\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)},\frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\right)\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}\right)^2 \qquad (2)$$

$$=\sum_{(x_i,y_i)\in W}\left(\left(\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)},\frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\right)\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}\right)^2$$

$$=\sum_{(x_i,y_i)\in W}(\Delta x,\Delta y)\begin{pmatrix}\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)}\\ \frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\end{pmatrix}\left(\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)},\frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\right)\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}$$

$$=(\Delta x,\Delta y)\left\{\sum_{(x_i,y_i)\in W}\begin{pmatrix}\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)}\\ \frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\end{pmatrix}\left(\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)},\frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\right)\right\}\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}$$

$$\boldsymbol{M}_{2\times 2}$$

$$=(\Delta x,\Delta y)\begin{pmatrix}\sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)}\right)^2 & \sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)}\right)\left(\frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\right)\\ \sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial x}\Big|_{(x_i,y_i)}\right)\left(\frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\right) & \sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial y}\Big|_{(x_i,y_i)}\right)^2\end{pmatrix}\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}$$

Harris feature—point detector (math)

$$s_W(\triangle x,\triangle y) \cong [\triangle x,\triangle y]\boldsymbol{M}\begin{bmatrix}\triangle x\\ \triangle y\end{bmatrix},$$

where, $\boldsymbol{M} = \begin{pmatrix} \sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial x}|_{(x_i,y_i)}\right)^2 & \sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial x}|_{(x_i,y_i)}\right)\left(\frac{\partial f}{\partial y}|_{(x_i,y_i)}\right) \\ \sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial x}|_{(x_i,y_i)}\right)\left(\frac{\partial f}{\partial y}|_{(x_i,y_i)}\right) & \sum_{(x_i,y_i)\in W}\left(\frac{\partial f}{\partial y}|_{(x_i,y_i)}\right)^2 \end{pmatrix}$

$s_W(\triangle x,\triangle y)=1$ represents the trace of $(\triangle x,\triangle y)$, making the appearance change of the local patch W centered at $\boldsymbol{x}$ equal to 1

When the position $\boldsymbol{x}$ and the window W are given, $\boldsymbol{M}$ is determined and the trace determined by the equation $s_W(\triangle x,\triangle y)=1$ is actually totally determined by $\boldsymbol{M}$

Harris feature—point detector (math)

Assignment

(1) $\boldsymbol{M}$ can be proved to be a positive semi-definite matrix

(2) In practice, $\boldsymbol{M}$ is positive definite nearly for sure, then $[\Delta x\ \Delta y]\boldsymbol{M}\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}=1$ represents an ellipse; the length of its semi-major axis is $1/\sqrt{\lambda_2}$ and the length of its semi-minor axis is $1/\sqrt{\lambda_1}$, where $\lambda_1 \geq \lambda_2 > 0$ are the two eigen-values of $\boldsymbol{M}$

(Hints: for problem (1), read appendix H; for problem (2), read appendix B)

The shape of such an ellipse is totally determined by $\boldsymbol{M}$

Harris feature—point detector (math)

Diagonalization of $\boldsymbol{M}$: $$\boldsymbol{M} = \boldsymbol{R}^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}\boldsymbol{R}$$

where $\boldsymbol{R}$ is an orthogonal matrix

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by $\boldsymbol{R}$

Harris feature—point detector (math)

The "cornerness" of the window W is reflected by $\boldsymbol{M}$

Suppose there are two local windows W_1 and W_2; consider the cases when the moving of the two windows leads to the intensity change equals to 1. The moving vector $[\Delta x, \Delta y]$ of each window satisfies the ellipse equation. Thus,

For W_1,

$$[\Delta x, \Delta y]\boldsymbol{M}_1\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}=1$$

Which window has higher cornerness?

For W_2,

$$[\Delta x, \Delta y]\boldsymbol{M}_2\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}=1$$

Harris feature—point detector (math)

Classification of image points using eigenvalues of $\boldsymbol{M}$:

Harris feature—point detector (cornerness measure)

Conducting EVD is not implementation efficient, and accordingly, Harris proposed an empirical cornerness measure

Measure of corner response at $\boldsymbol{x}$:

$$r(\boldsymbol{x}) = \det(\boldsymbol{M}(\boldsymbol{x})) - k\left(trace(\boldsymbol{M}(\boldsymbol{x}))\right)^2$$

where,

$$\det(\boldsymbol{M}(\boldsymbol{x})) = \lambda_1\lambda_2, trace(\boldsymbol{M}(\boldsymbol{x})) = \lambda_1 + \lambda_2$$

λ_1 and λ_2 are the two eigen-values of $\boldsymbol{M}(\boldsymbol{x})$

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

Harris feature—point detector (illustration)

Ellipse with equation : $[\triangle x, \triangle y]\boldsymbol{M}\begin{bmatrix}\triangle x\\ \triangle y\end{bmatrix}=1$

flat region
both eigenvalues small

edge
one small, one large

Harris feature—point detector (illustration)

Ellipse with equation : $[\triangle x, \triangle y]\boldsymbol{M}\begin{bmatrix}\triangle x\\ \triangle y\end{bmatrix}=1$

corner
both eigenvalues large

Harris feature—point detector (implementation)

where $g(x,y;\sigma)=\dfrac{1}{2\pi\sigma^2}\exp\left(-\dfrac{(x^2+y^2)}{2\sigma^2}\right)$ is a 2D Gaussian window, with the side-length 6σ

Harris feature—point detector (invariance)

- From the definition of the corner, theoretically, Harris corner detector is rotation invariant
- **However, it is not invariant to scale change!**

All points will be classified as edges

Corner !

The underlying reason is that Harris corner detection scheme does not provide an automatic and appropriate window size selection method!

Harris feature—feature descriptor

- "Block" descriptor for a Harris corner $\boldsymbol{x}$
  - Take a region with a fixed size around $\boldsymbol{x}$
  - Stack the region into a vector and normalize it as $\boldsymbol{d}$
  - This vector $\boldsymbol{d}$ serves as the descriptor for $\boldsymbol{x}$

Harris feature—feature descriptor

- "Block" descriptor for a Harris corner $\boldsymbol{x}$
  - Take a region with a fixed size around $\boldsymbol{x}$
  - Stack the region into a vector and normalize it as $\boldsymbol{d}$
  - This vector $\boldsymbol{d}$ serves as the descriptor for $\boldsymbol{x}$
- Deficiencies of such simple descriptors
  - Not rotation invariant
  - Not scale invariant

Harris feature—feature descriptor

- "Block" descriptor for a Harris corner $\boldsymbol{x}$
  - Take a region with a fixed size around $\boldsymbol{x}$
  - Stack the region into a vector and normalize it as $\boldsymbol{d}$
  - This vector $\boldsymbol{d}$ serves as the descriptor for $\boldsymbol{x}$
- We want:
  - **Rotation and scale invariant feature points**
  - **Rotation and scale invariant feature descriptors**

That motivates us to seek more advanced feature point detectors and feature point descriptors

Harris feature—distance between two descriptors

Given two (block) descriptors $\boldsymbol{d}_1, \boldsymbol{d}_2 \in \mathbb{R}^n$, their distance can be computed in different ways

(1) Sum of squared differences (SSD): $SSD_{dist}\left(\boldsymbol{d}_1, \boldsymbol{d}_2\right) = \left\|\boldsymbol{d}_1 - \boldsymbol{d}_2\right\|_2^2 = \sum_{i=1}^{n}\left(d_1^i - d_2^i\right)^2$

(2) Sum of absolute differences (SAD): $SAD_{dist}\left(\boldsymbol{d}_1, \boldsymbol{d}_2\right) = \sum_{i=1}^{n}\left|d_1^i - d_2^i\right|$

(3) Normalized cross correlation (NCC):

$$NCC_{dist}\left(\boldsymbol{d}_1, \boldsymbol{d}_2\right) = 1 - \frac{1}{n}\frac{\left(\boldsymbol{d}_1 - \mu\left(\boldsymbol{d}_1\right)\right)\cdot\left(\boldsymbol{d}_2 - \mu\left(\boldsymbol{d}_2\right)\right)}{std\left(\boldsymbol{d}_1\right)std\left(\boldsymbol{d}_2\right)}$$

or

$$NCC_{dist}\left(\boldsymbol{d}_1, \boldsymbol{d}_2\right) = \arccos\left(\frac{1}{n}\frac{\left(\boldsymbol{d}_1 - \mu\left(\boldsymbol{d}_1\right)\right)\cdot\left(\boldsymbol{d}_2 - \mu\left(\boldsymbol{d}_2\right)\right)}{std\left(\boldsymbol{d}_1\right)std\left(\boldsymbol{d}_2\right)}\right)$$

where $\mu\left(\boldsymbol{d}_1\right)$ returns the mean value of $\boldsymbol{d}_1$ and $std(\boldsymbol{d}_1)$ returns the standard deviation of $\boldsymbol{d}_1$

Content

- General requirements for interest point detectors and descriptors
- Harris feature
  - Point detector
  - Feature descriptor
  - Distance between two descriptors
- SIFT feature
  - Point detector
  - Feature descriptor
  - Distance between two descriptors
- ORB feature
  - Point detector
  - Feature descriptor
  - Distance between two descriptors
- Feature point matching
- Practice

SIFT feature—overview

- Deficiencies of the Harris feature
  - Harris corner detector is not scale invariant since it does not have an automatic size determination strategy for the local window W
  - "block" descriptor is not scale invariant since the size of its local window is set empirically

⬇

We expect **scale invariant feature points** and **scale invariant feature descriptors**

⬇

- SIFT feature can meet our needs
  - ✓ It has a scale invariant feature point detection scheme
  - ✓ It has a scale invariant feature descriptor construction scheme

SIFT feature—overview

- Point detector in SIFT
  - It uses a series of scale-normalized LoGs with different scales to detect blobs in the given image in a scale-invariant manner
  - When a blob is detected, a byproduct is also obtained, its **characteristic scale (scale covariant)**, which will be used to build the feature descriptor for the blob center
- Feature descriptor in SIFT
  - For a blob center, its feature descriptor is constructed based on its characteristic scale and thus it is scale-invariant

SIFT feature—overview

Prof. David Lowe. From 2015-2018, David Lowe was a Senior Research Scientist with Google in the Machine Intelligence Group. From 2009-2015 he was co-founder and chairman of Cloudburst Research, a computer vision startup in Vancouver that was acquired by Google in May 2015. From 1987-2015, he was a professor of computer science at the University of British Columbia

D.G. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60 (2), pp. 91-110, 2004

This paper has been cited by 78756 times by the day May 19, 2025

SIFT feature—point detector (basic idea)

- In SIFT, the feature point is the “blob center”
  - To characterize a blob, we need to know its center position and also its spatial extension (or its size)

SIFT feature—point detector (basic idea)

Tony Linderberg proposes to use scale-normalized LoG to detect blobs[1]

2D isotropic Gaussian function, $g(x,y)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$

LoG (Laplacian of Gaussian), $\nabla^2 g=\frac{\partial^2 g}{\partial x^2}+\frac{\partial^2 g}{\partial y^2}=\frac{x^2+y^2-2\sigma^2}{2\pi\sigma^6}e^{-\frac{x^2+y^2}{2\sigma^2}}$

Scale-normalized LoG,

$$\sigma^2\nabla^2 g=\frac{x^2+y^2-2\sigma^2}{2\pi\sigma^4}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

➡ It is quite suitable for **blob** structure detection

[1] T. Linderberg, Scale-space theory: A basic tool for analysing structures at different scales, J. Applied Statistics, 1994, 21(2): 224-270

SIFT feature—point detector (basic idea)

- For blob detection, we use a “template matching” scheme
- For a real blob point $\boldsymbol{x}$, its responses to scale-normalized LoGs with various scales has a **unique peak** (valley) , and thus we can use the “peak scale” to determine its size, that is, its **characteristic scale**

Responses of a blob point to a series of scale-normalized LoGs with various scales

SIFT feature—point detector (basic idea)

The responses of image $\boldsymbol{I}$ at $\boldsymbol{x}$ to the scale-normalized LoGs $\left\{\sigma_i^2\nabla^2 g\left(\sigma_i\right)\right\}_{i=1}^{n}$ are $\left\{\left(\sigma_i^2\nabla^2 g\left(\sigma_i\right)*\boldsymbol{I}\right)\left(\boldsymbol{x}\right)\right\}_{i=1}^{n}$

Let $\sigma_{\boldsymbol{x}}^*$ be,

$$\sigma_{\boldsymbol{x}}^* = \arg\max_{\sigma_i}\left(\sigma_i^2\nabla^2 g\left(\sigma_i\right)*\boldsymbol{I}\right)\left(\boldsymbol{x}\right), i = 1,\ldots,n$$

If $r\left(\boldsymbol{x}\right) \triangleq \left(\sigma_{\boldsymbol{x}}^2\nabla^2 g\left(\sigma_{\boldsymbol{x}}\right)*\boldsymbol{I}\right)\left(\boldsymbol{x}\right)$ is a unique peak and $\forall \boldsymbol{y} \in \boldsymbol{x}$'s neighborhood,

$$r\left(\boldsymbol{x}\right) > r\left(\boldsymbol{y}\right)$$

Then we say $\boldsymbol{x}$ is a feature point (blob center) and $\sigma_{\boldsymbol{x}}^*$ is its characteristic scale

Note: for the above statements, changing “peak”, “argmax”, and “>” to “valley”, “argmin”, and “<” also holds

SIFT feature—point detector (characteristic scale)

For a blob, its characteristic scale defined above is **scale-covariant**

An example

$\times 2$

x's characteristic scale obtained by scale-normalized LoGs is 1.5

y's characteristic scale obtained by scale-normalized LoGs is 3.0

scale-normalized LoG satisfy our requirement of automatic scale selection for building **scale-invariant feature descriptors**

SIFT feature—point detector (implementation)

- Interest points:
  - Local extremum in scale space of scale-normalized Laplacian-of-Gaussian and be greater than a threshold

$$\sigma^2\nabla^2 g(\sigma)$$

σ_5

σ_4

σ_3

σ_2

σ_1

SIFT feature—point detector (implementation)

- Interest points:
  - Local extremum in scale space of scale-normalized Laplacian-of-Gaussian and be greater than a threshold

SIFT feature—point detector (implementation)

- Interest points:
  - Local extremum in scale space of scale-normalized Laplacian-of-Gaussian and be greater than a threshold

SIFT feature—point detector (implementation)

- Interest points:
  - Local extremum in scale space of scale-normalized Laplacian-of-Gaussian and be greater than a threshold

$$\sigma^2\nabla^2 g(\sigma)$$

$\Rightarrow$ **List of** (x, y, σ)

(Positions of extrema in the scale-spatial space)

SIFT feature—point detector (implementation)

Note: local extrema is obtained as key points by comparing the examined location with all the other 26 points around it in the scale-space

If the local extrema of scale-normalized LoG is achieved at $\boldsymbol{p}$, two things of $\boldsymbol{p}$ can be determined: its spatial location and characteristic scale

SIFT feature—point detector (example)

SIFT feature—point detector (example)

sigma = 11.9912

SIFT feature—point detector (example)

SIFT feature—point detector (example)

SIFT feature—feature descriptor

- Scale Invariant Feature Transform
  - Proposed in [1]
  - It uses extrema of DoG to detect key points and the associated characteristic scales
  - It uses SIFT to describe a key point

[1] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60 (2), pp. 91-110, 2004

Prof. David Lowe
University of British Columbia

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

- Scan each DOG image
  - Look at all neighboring points (including scale)
  - Identify Min and Max
    - 26 Comparisons

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

Assign Keypoints Orientations

- Assign orientation to the keypoint
  - Find local orientation: dominant orientation of gradient for the image patch (its size is determined by the characteristic scale)

  - Rotate the patch according to this angle; this can achieve rotation invariance description

Assign Keypoints Orientations

- Orientation normalization
  - Compute orientation histogram
  - Select dominant orientation
  - Normalization: rotate the patch to the selected orientation

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

- Building the descriptor
  - Sample the points around the keypoint
  - Rotate the gradients and coordinates by the previously computed orientation
  - Separate the region in to 4×4 sub-regions
  - Create gradient-orientation histogram for each sub-region with 8 bins (In real implementation, each sample point is weighted by a Gaussian)

SIFT

- Building the descriptor

- Actual implementation uses 4*4 sub regions which lead to a 4*4*8 = 128 element vector

SIFT

- One image yields:
  - n 128-dimensional descriptors: each one is a histogram of the gradient orientations within a patch
    - [n x 128 matrix]
  - n scale parameters specifying the size of each patch
    - [n x 1 vector]
  - n orientation parameters specifying the angle of the patch
    - [n x 1 vector]
  - n 2D points giving positions of the patches
    - [n x 2 matrix]

Note: in the lecture, we simply introduce the general ideas of SIFT. For more efficient and accurate implementation, Lowe suggested many tricks, such as using DoG to approximate LoG, performing detection in the DoG pyramid, refining key point locations, filtering out edge points, etc. If you have interests in these details, you can refer to the textbook and the example implementation

SIFT

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

Applications of SIFT

- Object recognition
- Robot localization and mapping
- Panorama stitching
- 3D scene modeling, recognition and tracking
- Analyzing the human brain in 3D magnetic resonance images

Content

- Local Invariant Features
- Harris Corner Detector
- Scale Invariant Point Detection
- Scale Invariant Feature Transform (SIFT)
- Interest Point Matching

Distance between two descriptors (not limited to block descriptors)

Given two descriptors $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{R}^n$, their distance can be computed in different ways

Sum of squared differences (SSD):

$$SSD_{dist}\left(\mathbf{d}_1, \mathbf{d}_2\right) = \left\|\mathbf{d}_1 - \mathbf{d}_2\right\|_2^2 = \sum_{i=1}^{n}\left(d_1^i - d_2^i\right)^2$$

Or sum of absolute differences (SAD):

$$SAD_{dist}\left(\mathbf{d}_1, \mathbf{d}_2\right) = \sum_{i=1}^{n}\left|d_1^i - d_2^i\right|$$

Or normalized cross correlation (NCC):

$$NCC_{dist}\left(\mathbf{d}_1, \mathbf{d}_2\right) = 1 - \frac{1}{n}\frac{\left(\mathbf{d}_1 - \mu\left(\mathbf{d}_1\right)\right)\cdot\left(\mathbf{d}_2 - \mu\left(\mathbf{d}_2\right)\right)}{std\left(\mathbf{d}_1\right)std\left(\mathbf{d}_2\right)}$$

or

$$NCC_{dist}\left(\mathbf{d}_1, \mathbf{d}_2\right) = \arccos\left(\frac{1}{n}\frac{\left(\mathbf{d}_1 - \mu\left(\mathbf{d}_1\right)\right)\cdot\left(\mathbf{d}_2 - \mu\left(\mathbf{d}_2\right)\right)}{std\left(\mathbf{d}_1\right)std\left(\mathbf{d}_2\right)}\right)$$

where $\mu\left(\mathbf{d}_1\right)$ returns the mean value of $\mathbf{d}_1$ and $std(\mathbf{d}_1)$ returns the standard deviation of $\mathbf{d}_1$

Key points matching based on their descriptors

Suppose I_1 and I_2 are two images

I_1's key points and the associated descriptors are $\{\mathbf{x}_i\}_{i=1}^{m}$ and $\mathcal{P}=\{\mathbf{d}_i\}_{i=1}^{m}$

I_2's key points and the associated descriptors are $\{\mathbf{y}_j\}_{j=1}^{n}$ and $\mathcal{Q}=\{\mathbf{e}_j\}_{j=1}^{n}$

If and only if $\mathbf{x}_i$' and $\mathbf{y}_j$'s descriptors $\mathbf{d}_i$ and $\mathbf{e}_j$ satisfy the following conditions, we say the key points $\mathbf{x}_i$ and $\mathbf{y}_j$ match and form a correspondence pair,

1) $dist(\mathbf{d}_i, \mathbf{e}_j)<t_1$, where t_1 is a predefined threshold

2) $\mathbf{d}_i$ and $\mathbf{e}_j$ satisify the "two-direction" confirmation criteria, i.e.,

$$\forall \mathbf{v} \in \mathcal{P}, \mathbf{v} \neq \mathbf{d}_i, dist\left(\mathbf{v}, \mathbf{e}_j\right) > dist\left(\mathbf{d}_i, \mathbf{e}_j\right)$$

$$\forall \mathbf{v} \in \mathcal{Q}, \mathbf{v} \neq \mathbf{e}_j, dist\left(\mathbf{d}_i, \mathbf{v}\right) > dist\left(\mathbf{d}_i, \mathbf{e}_j\right)$$

3) $\mathbf{d}_i$ and $\mathbf{e}_j$'s matching is unambiguous

Let $d_1 = dist\left(\mathbf{d}_i, \mathbf{e}_j\right)$. Suppose $\mathbf{e}_k$ is the second best matching descriptor to $\mathbf{d}_i$ and let $d_2 = dist\left(\mathbf{d}_i, \mathbf{e}_k\right)$. Then, $d_1 / d_2 < t_2$, where t_2 is another predefined threshold

Key points matching based on their descriptors

Matching Harris corners using block descriptors

Key points matching based on their descriptors

Matching scale-invariant points using SIFT descriptors

Key points matching based on their descriptors

Key points matching based on their descriptors

Matching scale-invariant points using SIFT descriptors

Q+A