Overview

- Motivation of Work
- Overview of Algorithm
- Scale Space and Difference of Gaussian
- Keypoint Localization
- Orientation Assignment
- Descriptor Building
- Application
Motivation of Work

- Image Matching
- Correspondence Problem
- Desirable Feature Characteristics
- Scale Invariance
- Rotation Invariance
- Illumination invariance
- Viewpoint invariance
Why do we care about matching features?

- Object Recognition
- Tracking/SFM
We want invariance!!!

- Good features should be robust to all sorts of nastiness that can occur between images.

Types of invariance

- Illumination
Types of invariance

- Illumination
- Scale
Types of invariance

- Illumination
- Scale
- Rotation
Types of invariance

- Illumination
- Scale
- Rotation
- Affine
Types of invariance

- Illumination
- Scale
- Rotation
- Affine
- Full Perspective
How to achieve illumination invariance

- The easy way (normalized)
- Difference based metrics (random tree, Haar, and sift)
Algorithm Overview

1. Construct Scale Space
2. Take Difference of Gaussians
3. Locate DoG Extrema
4. Sub Pixel Locate Potential Feature Points
5. Filter Edge and Low Contrast Responses
6. Assign Keypoints Orientations
7. Build Keypoint Descriptors
8. Go Play with Your Features!!
Constructing Scale Space

- 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!!
How to achieve scale invariance

• Pyramids
  • Divide width and height by 2
  • Take average of 4 pixels for each pixel (or Gaussian blur)
  • Repeat until image is tiny
  • Run filter over each size image and hope its robust

• Scale Space (DOG method)
Pyramids
How to achieve scale invariance

- Pyramids
- Scale Space (DOG method)
  - Like having a nice linear scaling without the expense
  - Take features from differences of these images
  - If the feature is repeatably present in between Difference of Gaussians it is Scale Invariant and we should keep it.
Constructing Scale Space

- Gaussian kernel used to create scale space
- Only possible scale space kernel (Lindberg 94)

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

where

\[ G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \]
Laplacian of Gaussians

- LoG - $\sigma^2\Delta^2 G$
- Extrema Useful
  - Found to be stable features
  - Gives Excellent notion of scale
- Calculation costly so instead....
Differences Of Gaussians
DoG Pyramid
The top line of the first graph shows the percent of keypoints that are repeatably detected at the same location and scale in a transformed image as a function of the number of scales sampled per octave. The lower line shows the percent of keypoints that have their descriptors correctly matched to a large database. The second graph shows the total number of keypoints detected in a typical image as a function of the number of scale samples.
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!!
Locate the Extrema of the DoG

- Scan each DOG image
- Look at all neighboring points (including scale)
- Identify Min and Max
Sub pixel Localization

1. Construct Scale Space
2. Take Difference of Gaussians
3. Locate DoG Extrema
4. Sub Pixel Locate Potential Feature Points
5. Filter Edge and Low Contrast Responses
6. Assign Keypoints Orientations
7. Build Keypoint Descriptors
8. Go Play with Your Features!!
Sub pixel Localization

- 3D Curve Fitting
- Taylor Series Expansion

\[ D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \]

- Differentiate and set to 0

\[ \hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}. \]

- to get location in terms of \((x, y, \sigma)\)
Filter Low Contrast Points

- Low Contrast Points Filter
- Use Scale Space value at previously found location

\[ D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \hat{x}} \hat{x}. \]
The House With Contrast Elimination
Edge Response Elimination

- Peak has high response along edge, poor other direction

- A poorly defined peak in the difference-of-Gaussian function will have a large principal curvature across the edge but a small one in the perpendicular direction. The principal curvatures can be computed from a 2x2 Hessian matrix.

- Use Hessian
  - Eigenvalues Proportional to principle Curvatures
  - Use Trace and Determinant

\[
\text{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta, \quad \text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \\
\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r+1)^2}{r}
\]
Results On The House

Apply Contrast Limit

Apply Contrast and Edge Response Elimination
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!!
Orientation Assignment

- Compute Gradient for each blurred image

\[
m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}
\]
\[
\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1))/(L(x + 1, y) - L(x - 1, y)))
\]

- For region around keypoint
  - Create Histogram with 36 bins for orientation
  - Weight each point with Gaussian window of $1.5\sigma$
  - Create keypoint for all peaks with value $\geq .8$ max bin
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!!
Building the Descriptor

- Find the blurred image of closest scale
- Sample the points around the keypoint
- Rotate the gradients and coordinates by the previously computer orientation
- Separate the region into sub regions
- Create histogram for each sub region with 8 bins
  - Weight the samples with $N(\sigma) = 1.5$ Region width
Actual implementation uses $4 \times 4$ descriptors from $16 \times 16$ which leads to a $4 \times 4 \times 8 = 128$ element vector.
Results check

- Scale Invariance
- Scale Space usage – Check
- Rotation Invariance
- Align with largest gradient – Check
- Illumination Invariance
- Normalization – Check
- Viewpoint Invariance
- For small viewpoint changes – Check (mostly)
Results
Results
Questions?
Credits

- Pele, Ofir. SIFT: Scale Invariant Feature Transform. Sift.ppt