CAP4453-Robot Vision
Lecture 8-Scale Invariant Feature Transform (SIFT)

Ulas Bagci
bagci@ucf.edu
Outline

• Concept of Scale
  – Pyramids
  – Scale-space approaches briefly
• Scale invariant region selection
• **SIFT**: an image region descriptor

• **David Lowe, IJCV 2004 paper** [Please read it!]
Reminder: Motivation

• Image Matching
  – Fundamental aspect of many problems
    • Object Recognition
    • 3D Structures
    • Stereo Correspondence
    • Motion Tracking
    • ....

What are the desired features to conduct these tasks?
Review of the Corner Detection

Corners are invariant to:
- Translation: ✔
- Rotation: ✔
- Scaling: ✗
Scale Invariant Feature Transform (SIFT)

• Lowe., D. 2004, IJCV

Distinctive Image Features from Scale-Invariant Keypoints

DAVID G. LOWE
Computer Science Department, University of British Columbia, Vancouver, B.C., Canada
Lowe@cs.ubc.ca

Received January 10, 2003; Revised January 7, 2004; Accepted January 22, 2004

Abstract. This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

Keywords: invariant features, object recognition, scale invariance, image matching

1. Introduction

Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and motion tracking. This paper describes image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. Large numbers of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition.

The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test. Following are the major stages of computation used to generate the set of image features:

1. Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All further operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
Properties of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint
- Can handle significant changes in illumination
- Efficient (real time)
- Lots of code available
Scale Invariant Feature Transform (SIFT)

• Lowe., D. 2004, IJCV

Abstract. This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

Keywords: invariant features, object recognition, scale invariance, image matching

1. Introduction

Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and motion tracking. This paper describes image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. Large numbers of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition.

The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test. Following are the major stages of computation used to generate the set of image features:

1. Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
Scale-Space

• Any given image can contain objects that exist at scales different from other objects in the same image
• Or, that even exist at multiple scales simultaneously
• Neighborhood operations can only extract local features (at a scale of at most a few pixels), however, images contain information at larger scales
Multi-scale Representation

Image

Small Scale Features

Smaller Filter Masks

Large Scale Features

Larger Filter Masks

Computational Cost increases!

Doubling the scale leads to four-fold increase in the number of operations in 2D.
Multi-Scale Representation: Pyramid

- **Pyramid** is one way to represent images in multi-scale
  - Pyramid is built by using multiple copies of image.
  - Each level in the pyramid is 1/4 of the size of previous level.
  - The lowest level is of the highest resolution.
  - The highest level is of the lowest resolution.
Multi-Scale Representation: Pyramid
Pyramid can capture global and local features
Gaussian Pyramids (=blur + subsample)

| 512 | 256 | 128 | 64 | 32 | 16 | 8 |

Source: Forsyth
Decimation and Interpolation

LowPass filter (i.e., Gaussian)
Scale Selection-Automated

- Find scale that gives local maxima of some function $f$ in both position and scale.
Good Function?

• A “good” function for scale detection: has one stable sharp peak

• For usual images: a good function would be a one which responds to contrast (sharp local intensity change)
Patch Size Corresponding to Scale

How to find corresponding patch sizes?

\[ f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I_{i_1 \ldots i_m}(x', \sigma')) \]
Patch Size Corresponding to Scale

\[ f(I_{i_1\ldots i_m}(x, \sigma)) \]

\[ f(I_{i_1\ldots i_m}(x', \sigma)) \]
Patch Size Corresponding to Scale

\[ f(I_{i_1 \ldots i_m}(x, \sigma)) \]

\[ f(I_{i_1 \ldots i_m}(x', \sigma)) \]
Patch Size Corresponding to Scale

\[ f(I_{i_1...i_n}(x, \sigma)) \]

\[ f(I_{i_1...i_m}(x', \sigma)) \]
Patch Size Corresponding to Scale

\[ f(I_{i_1...i_m}(x, \sigma)) \]

\[ f(I_{i_1...i_m}(x', \sigma)) \]
Patch Size Corresponding to Scale

\[ f(I_{i_1 \ldots i_m}(x, \sigma)) \]

\[ f(I_{i_1 \ldots i_m}(x', \sigma)) \]

K. Grauman, B. Leibe
Patch Size Corresponding to Scale

- Function responses for increasing scale (scale signature)
Normalization

Normalize: rescale to fixed size
Scale Invariant Feature Transform (SIFT)

- Shows existence, importance, and value of invariant types of detector
- Demonstrates the richness that feature descriptors can bring to feature matching

Instead of using LoG (Laplacian of Gaussian) like in Harris and Hessian-based operators, SIFT uses DoG (Difference of Gaussian).
Difference of Gaussian (DoG)

- It is a common approximation of LoG (better run time).

$$D_{\sigma,\alpha}(x, y) = L(x, y, \alpha) - L(x, y, \alpha\sigma)$$

- It is the difference between a blurred copy of image I and an even more blurred copy of I.

$$\nabla G_\sigma(x, y) \approx \frac{G_{\alpha\sigma}(x, y) - G_\sigma(x, y)}{(\alpha - 1)\sigma^2}$$
Scale Invariant Feature Transform (SIFT)

• Image content is transformed into **local feature coordinates** that are invariant to translation, rotation, scale, and other imaging parameters.
Overall Procedure of SIFT

Scale-Space Extrema Detection
Search over multiple scales and image locations

KeyPoint Localization
Fit a model to determine location and scale. Select KeyPoints based on a measure of stability.

Orientation Assignment
Compute best orientation(s) for each keyPoint region.

KeyPoint Description
Use local image gradients at selected scale and rotation to describe each keyPoint region.
SIFT Descriptor

Basic idea:

- Take n x n (i.e., n=16) square window (around a feature/interest point)
- Divide them into m x m (i.e., m=4) cells
- Compute gradient orientation for each cell
- Create histogram over edge orientations weighted by magnitude

![Image](image-gradients.png)

![Histogram](angle-histogram.png)
SIFT Descriptor

16 histograms x 8 orientations = 128 features
SIFT Descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor
- Threshold normalize the descriptor:

\[ \sum_{i} d_{i}^{2} = 1 \quad \text{such that:} \quad d_{i} < 0.2 \]
Revisit SIFT Steps

(1) Scale-space extrema detection
   – Extract scale and rotation invariant interest points (i.e., keypoints).

(2) KeyPoint localization
   – Determine location and scale for each interest point.
   – Eliminate “weak” keypoints

(3) Orientation assignment
   – Assign one or more orientations to each keypoint.

(4) KeyPoint descriptor
   – Use local image gradients at the selected scale.
(1) Scale-Space Extrema Detection

- We want to find points that give us information about the objects in the image.

- The information about the objects is in the object’s edges.

- We will represent the image in a way that gives us these edges as this representations extrema points.
(1) Scale-Space Extrema Detection

- **Harris-Laplace**
  
  *Find local maxima of:*
  - Harris detector in space
  - LoG in scale

- **SIFT**
  
  *Find local maxima of:*
  - Hessian in space
  - DoG in scale
(1) Scale-Space Extrema Detection
(1) Scale-Space Extrema Detection

- Extract local extrema (i.e., minima or maxima) in DoG pyramid.
  - Compare each point to its 8 neighbors at the same level, 9 neighbors in the level above, and 9 neighbors in the level below (i.e., 26 total).
(2) KeyPoint Localization

• There are still a lot of points, some of them are not good enough.
• The locations of keypoints may be not accurate.
• Eliminating edge points.
(2) KeyPoint Localization

- Reject (1) points with low contrast (flat)
  (2) poorly localized along an edge (edge)
(2) KeyPoint Localization

- Reject (1) points with low contrast (flat)
- (2) poorly localized along an edge (edge)

\[
\begin{align*}
\text{if} & \quad \frac{Tr(H)}{|H|} < \frac{(r + 1)^2}{r} \\
\text{Reject} & \\
\text{Where} & \quad r = \frac{\lambda_1}{\lambda_2} \quad \text{and practically SIFT uses } r=10.
\end{align*}
\]
Inaccurate KeyPoint Localization

- Poor contrast
Inaccurate KeyPoint Localization

The Solution:

Taylor expansion:

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

Minimize to find accurate extrema:

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

If offset from sampling point is larger than 0.5 - Keypoint should be in a different sampling point.

Brown & Lowe 2002
Local extremas

After removing low contrast features
Local extremas

After removing poorly localized edges
SIFT Descriptor
(3) Orientation Assignment

Create histogram of gradient directions, within a region around the keyPoint, at selected scale:

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

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

\[ \theta(x,y) = a \tan 2( (L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \]

- Histogram entries are weighted by (i) gradient magnitude and (ii) a Gaussian function with \( \sigma \) equal to 1.5 times the scale of the keypoint.

36 bins (i.e., 10° per bin)
(3) Orientation Assignment

- Keypoint location = extrema location
- Keypoint scale is scale of the DOG image
(3) Orientation Assignment

gradient magnitude

weighted by 2D gaussian kernel

weighted gradient magnitude
(3) Orientation Assignment

Weighted gradient magnitude

Weighted orientation histogram. Each bucket contains sum of weighted gradient magnitudes corresponding to angles that fall within that bucket.

36 buckets
10 degree range of angles in each bucket, i.e.
- $0 \leq \text{ang} < 10$: bucket 1
- $10 \leq \text{ang} < 20$: bucket 2
- $20 \leq \text{ang} < 30$: bucket 3 ...
(3) Orientation Assignment

Orientation of keypoint is approximately 25 degrees
Is SIFT (exactly) Rotation invariant?

short answer: No (large rotations will distort rotation invariancy)

But then, why we look at the orientation histograms then?

for small rotations, SIFT will capture the feature points that we seek for!

small rotations → distribution of orientation (i.e., histogram) has the same statistics!

RIFT is a GENERALIZED version of SIFT with FULL rotation invariant property
both small and large rotations are kept!
RIFT: Rotation Invariant Feature Transform

(3) Orientation Assignment

1) You can assign the most dominant orientation to feature point (from histogram of orientations) as shown below

2) If there are more than 1 high value, you can use both of them

3) Or, you can use full spectrum of orientation vector (in the histogram)
(4) KeyPoint Descriptor

- **Partial Voting**: distribute histogram entries into adjacent bins (i.e., *additional robustness to shifts*)
  - Each entry is added to all bins, multiplied by a weight of $1-d$, where $d$ is the distance from the bin it belongs.
(4) KeyPoint Descriptor

- Descriptor depends on two main parameters:
  - (1) number of orientations
  - n x n array of orientation histograms
Change of Illumination

- **Change of brightness** => doesn’t effect gradients (difference of pixels value).
- **Change of contrast** => doesn’t effect gradients (up to normalization).
- **Saturation** (non-linear change of illumination) => affects magnitudes much more than orientation.
  => Threshold gradient magnitudes to 0.2 and renormalize.
Illumination invariance?

1. Before you start, you can normalize the images.
2. Difference based metrics can be used (Haar, SIFT,...)
• Extract features
• Extract features
• Compute *putative matches*
• Extract features
• Compute *putative matches*
• Loop:
  – *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
• Extract features
• Compute *putative matches*
• Loop:
  – *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
  – *Verify* transformation (search for other matches consistent with $T$)
• Extract features
• Compute *putative matches*
• Loop:
  – *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
  – *Verify* transformation (search for other matches consistent with $T$)
Object Recognition

- **For training images:**
  - Extracting keypoints by SIFT.
  - Creating descriptors database.

- **For query images:**
  - Extracting keypoints by SIFT.
  - For each descriptor - finding nearest neighbor in DB.
  - Finding cluster of at-least 3 keypoints.
  - Performing detailed geometric fit check for each cluster.
Image Registration
Questions ?