Lecture 3
Local Feature Descriptors and Matching

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2017
Content

- Scale Invariant Feature Transform
- Case Study: Homography Estimation
  - Matrix Differentiation
  - RANSAC-based Homography Estimation
- 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

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

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!!

Lin ZHANG, SSE, 2017
SIFT

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!!
SIFT

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

1. Construct Scale Space
2. Take Difference of Gaussians
3. Locate DoG Extrema
   - Sub Pixel Locate Potential Feature Points
4. Filter Edge and Low Contrast Responses
5. Assign Keypoints Orientations
6. 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!!
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

- Building the descriptor
  - Sample the points around the keypoint
  - Rotate the gradients and coordinates by the previously computed orientation
  - Separate the region into $4 \times 4$ 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
One image yields:

- $n$ 128-dimensional descriptors: each one is a histogram of the gradient orientations within a patch
  - $[n \times 128 \text{ matrix}]$
- $n$ scale parameters specifying the size of each patch
  - $[n \times 1 \text{ vector}]$
- $n$ orientation parameters specifying the angle of the patch
  - $[n \times 1 \text{ vector}]$
- $n$ 2D points giving positions of the patches
  - $[n \times 2 \text{ matrix}]$
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
Applications of SIFT

- Object recognition
Applications of SIFT

- Object recognition
Content

- Scale Invariant Feature Transform
- Case Study: Homography Estimation
  - Matrix Differentiation
  - RANSAC-based Homography Estimation
Matrix differentiation

- Function is a vector and the variable is a scalar

\[ f(t) = [f_1(t), f_2(t), \ldots, f_n(t)]^T \]

Definition

\[ \frac{df}{dt} = \begin{bmatrix} \frac{df_1(t)}{dt}, \ & \frac{df_2(t)}{dt}, & \cdots, & \frac{df_n(t)}{dt} \end{bmatrix}^T \]
Matrix differentiation

- Function is a matrix and the variable is a scalar

\[
f(t) = \begin{bmatrix}
  f_{11}(t) & f_{12}(t) & \cdots & f_{1m}(t) \\
  f_{21}(t) & f_{22}(t) & \cdots & f_{2m}(t) \\
  \vdots & \vdots & \ddots & \vdots \\
  f_{n1}(t) & f_{n2}(t) & \cdots & f_{nm}(t)
\end{bmatrix} = \begin{bmatrix}
  f_{ij}(t)
\end{bmatrix}_{n\times m}
\]

Definition

\[
\frac{df}{dt} = \begin{bmatrix}
  \frac{df_{11}}{dt} & \frac{df_{12}}{dt} & \cdots & \frac{df_{1m}}{dt} \\
  \frac{df_{21}}{dt} & \frac{df_{22}}{dt} & \cdots & \frac{df_{2m}}{dt} \\
  \vdots & \vdots & \ddots & \vdots \\
  \frac{df_{n1}}{dt} & \frac{df_{n2}}{dt} & \cdots & \frac{df_{nm}}{dt}
\end{bmatrix} = \begin{bmatrix}
  \frac{df_{ij}}{dt}
\end{bmatrix}_{n\times m}
\]
Matrix differentiation

- Function is a scalar and the variable is a vector

\[ f(\mathbf{x}), \mathbf{x} = (x_1, x_2, \ldots, x_n)^T \]

Definition

\[
\frac{df}{dx} = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right]^T
\]

In a similar way,

\[ f(\mathbf{x}), \mathbf{x} = (x_1, x_2, \ldots, x_n) \]

\[
\frac{df}{dx} = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right]
\]
Matrix differentiation

- Function is a vector and the variable is a vector
  \[ \mathbf{x} = [x_1, x_2, \ldots, x_n]^T, \mathbf{y} = [y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_m(\mathbf{x})]^T \]

Definition

\[
\frac{d\mathbf{y}}{d\mathbf{x}}^T = \begin{bmatrix}
\frac{\partial y_1(\mathbf{x})}{\partial x_1}, \frac{\partial y_1(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial y_1(\mathbf{x})}{\partial x_n} \\
\frac{\partial y_2(\mathbf{x})}{\partial x_1}, \frac{\partial y_2(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial y_2(\mathbf{x})}{\partial x_n} \\
\vdots \\
\frac{\partial y_m(\mathbf{x})}{\partial x_1}, \frac{\partial y_m(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial y_m(\mathbf{x})}{\partial x_n}
\end{bmatrix}_{m \times n}
\]
Matrix differentiation

• Function is a vector and the variable is a vector
  \[ \mathbf{x} = [x_1, x_2, \ldots, x_n]^T, \mathbf{y} = [y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_m(\mathbf{x})]^T \]

In a similar way,

\[
\frac{d\mathbf{y}^T}{d\mathbf{x}} = \begin{bmatrix}
\frac{\partial y_1(\mathbf{x})}{\partial x_1}, & \frac{\partial y_2(\mathbf{x})}{\partial x_1}, & \ldots, & \frac{\partial y_m(\mathbf{x})}{\partial x_1} \\
\frac{\partial y_1(\mathbf{x})}{\partial x_2}, & \frac{\partial y_2(\mathbf{x})}{\partial x_2}, & \ldots, & \frac{\partial y_m(\mathbf{x})}{\partial x_2} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial y_1(\mathbf{x})}{\partial x_n}, & \frac{\partial y_2(\mathbf{x})}{\partial x_n}, & \ldots, & \frac{\partial y_m(\mathbf{x})}{\partial x_n}
\end{bmatrix}_{n \times m}
\]
Matrix differentiation

- Function is a vector and the variable is a vector

Example:

\[ y = \begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad y_1(x) = x_1^2 - x_2, \quad y_2(x) = x_3^2 + 3x_2 \]

\[
\frac{dy^T}{dx} = \begin{bmatrix}
\frac{\partial y_1(x)}{\partial x_1} & \frac{\partial y_2(x)}{\partial x_1} \\
\frac{\partial y_1(x)}{\partial x_2} & \frac{\partial y_2(x)}{\partial x_2} \\
\frac{\partial y_1(x)}{\partial x_3} & \frac{\partial y_2(x)}{\partial x_3}
\end{bmatrix} = \begin{bmatrix}
2x_1 & 0 \\
-1 & 3 \\
0 & 2x_3
\end{bmatrix}
\]
Matrix differentiation

- Function is a scalar and the variable is a matrix

\[ f(X), X \in \mathbb{R}^{m \times n} \]

**Definition**

\[
\frac{df(X)}{dX} = \begin{bmatrix}
\frac{\partial f}{\partial x_{11}} & \frac{\partial f}{\partial x_{12}} & \ldots & \frac{\partial f}{\partial x_{1n}} \\
\frac{\partial f}{\partial x_{21}} & \frac{\partial f}{\partial x_{22}} & \ldots & \frac{\partial f}{\partial x_{2n}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f}{\partial x_{m1}} & \frac{\partial f}{\partial x_{m2}} & \ldots & \frac{\partial f}{\partial x_{mn}}
\end{bmatrix}
\]
Matrix differentiation

• Useful results

(1) \[ x, a \in \mathbb{R}^{n \times 1} \]

Then,

\[ \frac{da^T x}{dx} = a, \quad \frac{dx^T a}{dx} = a \]

How to prove?
Matrix differentiation

• Useful results

(2) \( A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^{n \times 1} \) Then, \( \frac{dAx}{dx^T} = A \)

(3) \( A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^{n \times 1} \) Then, \( \frac{dx^T A^T}{dx} = A^T \)

(4) \( A \in \mathbb{R}^{n \times n}, x \in \mathbb{R}^{n \times 1} \) Then, \( \frac{dx^T Ax}{dx} = (A + A^T)x \)

(5) \( X \in \mathbb{R}^{m \times n}, a \in \mathbb{R}^{m \times 1}, b \in \mathbb{R}^{n \times 1} \) Then, \( \frac{da^T X b}{dX} = ab^T \)

(6) \( X \in \mathbb{R}^{n \times m}, a \in \mathbb{R}^{m \times 1}, b \in \mathbb{R}^{n \times 1} \) Then, \( \frac{da^T X^T b}{dX} = ba^T \)

(7) \( x \in \mathbb{R}^{n \times 1} \) Then, \( \frac{dx^T x}{dx} = 2x \)
Content

• Scale Invariant Feature Transform
• Case Study: Homography Estimation
  • Matrix Differentiation
  • RANSAC-based Homography Estimation
Homography Estimation

Problem definition:
Given a set of points \( \{ \mathbf{x}_i \} \) and a corresponding set of points \( \{ \mathbf{x}'_i \} \) in a projective plane, compute the projective transformation that takes \( \mathbf{x}_i \) to \( \mathbf{x}'_i \).

We know there existing an \( H \) satisfying \( \mathbf{x}'_i = H \mathbf{x}_i, i = 1, 2, \ldots, n \).

Coordinates of \( \{ \mathbf{x}_i \} \) and \( \{ \mathbf{x}'_i \} \) are known, we need to find \( H \) where \( H \) is a homography matrix

\[
H = \begin{bmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{21} & a_{22} & a_{23} \\
    a_{31} & a_{32} & a_{33}
\end{bmatrix}
\]

It has 8 degrees of freedom
Homography Estimation

4 point-correspondence pairs can uniquely determine a homography matrix since each correspondence pair solves two degrees of freedom

\[
\begin{pmatrix}
cu \\
cv \\
c
\end{pmatrix} =
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\Rightarrow
\begin{cases}
a_{11}x + a_{12}y + a_{13} = cu \\
a_{21}x + a_{22}y + a_{23} = cv \\
a_{31}x + a_{32}y + a_{33} = c
\end{cases}
\]

\[
\begin{align*}
a_{11}x + a_{12}y + a_{13} &= u \\
a_{31}x + a_{32}y + a_{33} &= v
\end{align*}
\]
Homography Estimation

4 point-correspondence pairs can uniquely determine a homography matrix since each correspondence pair solves two degrees of freedom.

\[
\begin{pmatrix}
-x & -y & -1 & 0 & 0 & 0 & 0 & ux & uy & u \\
0 & 0 & 0 & -x & -y & -1 & 0 & vx & vy & v
\end{pmatrix}
\begin{pmatrix}
a_{11} \\
a_{12} \\
a_{13} \\
a_{21} \\
a_{22} \\
a_{23} \\
a_{31} \\
a_{32} \\
a_{33}
\end{pmatrix} = 0
\]

Thus, four correspondence pairs generate 8 equations.
Homography Estimation

4 point-correspondence pairs can uniquely determine a homography matrix since each correspondence pair solves two degrees of freedom

$$A\mathbf{x} = 0 \quad (1)$$

$8 \times 9 \quad 9 \times 1$

Normally, $\text{Rank}(A) = 8$; thus (1) has 1 (9-8) solution vector in its solution space.
Homography Estimation

• How about the case when there are more than 4 correspondence pairs?
  • Use the LS method to solve the model directly?
  • NO! Because usually, outliers exist among the correspondence pairs

RANdom SAmple Consensus (RANSAC) framework is a good candidate to solve this kind of issues
Homography Estimation

**Objective**
Robust fit a model to a data set $S$ which contains outliers

**Algorithm**

1. Randomly select a sample of $s$ data points from $S$ and instantiate the model from this subset
2. Determine the set of data points $S_i$ which are within a distance threshold $t$ of the model. The set $S_i$ is the consensus set of the sample and defines the inliers of $S$
3. If the size of $S_i$ (the number of inliers) is greater than some threshold $T$, re-estimate the model using all points in $S_i$ and terminate
4. If the size of $S_i$ is less than $T$, select a new subset and repeat the above
5. After $N$ trials the largest consensus set $S_i$ is selected, and the model is re-estimated using all points in the subset $S_i$
Homography Estimation

Line fitting: least square
Homography Estimation

Line fitting: RANSAC
Homography Estimation

Line fitting by RANSAC
Homography Estimation

Line fitting by RANSAC

- Randomly select two points
Homography Estimation

Line fitting by RANSAC

- Randomly select two points
- The hypothesized model is the line passing through the two points
Homography Estimation

Line fitting by RANSAC

- Randomly select two points
- The hypothesized model is the line passing through the two points
Line fitting by RANSAC

- Randomly select two points
- The hypothesized model is the line passing through the two points
Homography Estimation

Line fitting by RANSAC

- Test another two points
Homography Estimation

Line fitting by RANSAC

• The final fitting result
Homography Estimation: Example 1
Homography Estimation: Example 1

Interest points detection
Correspondence estimation

Then, the homography matrix can be estimated by using the correspondence pairs with RANSAC
Homography Estimation: Example 1

Transform image one using the estimated homography matrix
Finally, stitch the transformed image one with image two.
Homography Estimation: Example 2
Homography Estimation: Example 2

Interest points detection
Then, the homography matrix can be estimated by using the correspondence pairs with RANSAC.
Homography Estimation: Example 2

Transform image one using the estimated homography matrix
Finally, stitch the transformed image one with image two
Homography Estimation: Example 3

Project products of students from 2009 Media&Arts