Lecture 3
Local Feature Descriptors and Matching

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2021
Content

• Scale Invariant Feature Transform
• Case Study: Homography Estimation
  • Matrix Differentiation
  • Lagrange Multiplier
  • Least-squares for Linear Systems
  • RANSAC-based Homography Estimation
SIFT

• 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


Prof. David Lowe
University of British Columbia
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

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, 2021
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!!
Scan each DOG image
- Look at all neighboring points (including scale)
- Identify Min and Max
  - 26 Comparisons
SIFT

1. Construct Scale Space
   - Take Difference of Gaussians
     - Locate DoG Extrema
       - Sub Pixel Locate Potential Feature Points
2. 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 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
    - \([n \times 128\) matrix]

  ➢ *n* scale parameters specifying the size of each patch
    - \([n \times 1\) vector]

  ➢ *n* orientation parameters specifying the angle of the patch
    - \([n \times 1\) vector]

  ➢ *n* 2D points giving positions of the patches
    - \([n \times 2\) matrix]
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!!
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
  • Lagrange Multiplier
  • Least-squares for Linear Systems
  • RANSAC-based Homography Estimation
Matrix differentiation

• Function is a vector and the variable is a scalar

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

Definition

\[ \frac{df}{dt} = \left[ \frac{df_1(t)}{dt}, \frac{df_2(t)}{dt}, \ldots, \frac{df_n(t)}{dt} \right]^T \]
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

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

• Function is a scalar and the variable is a vector

\[ f(x), \quad x = (x_1, x_2, ..., x_n)^T \]

Definition

\[
\frac{df}{dx} = \begin{bmatrix}
\frac{\partial f}{\partial x_1}, & \frac{\partial f}{\partial x_2}, & ..., & \frac{\partial f}{\partial x_n}
\end{bmatrix}^T
\]

In a similar way,

\[ f(x), \quad x = (x_1, x_2, ..., x_n) \]

\[
\frac{df}{dx} = \begin{bmatrix}
\frac{\partial f}{\partial x_1}, & \frac{\partial f}{\partial x_2}, & ..., & \frac{\partial f}{\partial x_n}
\end{bmatrix}
\]
Matrix differentiation

• Function is a vector and the variable is a vector

\[ \mathbf{x} = \begin{bmatrix} x_1, x_2, \ldots, x_n \end{bmatrix}^T, \quad \mathbf{y} = \begin{bmatrix} y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_m(\mathbf{x}) \end{bmatrix}^T \]

Definition

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

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

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,

\[
\frac{d a^T x}{d x} = a, \quad \frac{dx^T a}{dx} = a
\]
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, \( \frac{dA^T}{dx} = A^T \)
  
  (4) \( A \in \mathbb{R}^{n \times n}, x \in \mathbb{R}^{n \times 1} \) Then, \( \frac{dA^TxAx}{dx} = (A + A^T)x \)
  
  (5) \( X \in \mathbb{R}^{m \times n}, a \in \mathbb{R}^{m \times 1}, b \in \mathbb{R}^{n \times 1} \) Then, \( \frac{da^TXb}{dX} = ab^T \)
  
  (6) \( X \in \mathbb{R}^{n \times m}, a \in \mathbb{R}^{m \times 1}, b \in \mathbb{R}^{n \times 1} \) Then, \( \frac{da^TX^Tb}{dX} = ba^T \)
  
  (7) \( x \in \mathbb{R}^{n \times 1} \) Then, \( \frac{dx^Tx}{dx} = 2x \)
Content

• Scale Invariant Feature Transform
• Case Study: Homography Estimation
  • Matrix Differentiation
  • Lagrange Multiplier
  • Least-squares for Linear Systems
  • RANSAC-based Homography Estimation
Lagrange multiplier

• Single-variable function

\( f(x) \) is differentiable in \((a, b)\). At \( x_0 \in (a, b) \), \( f(x) \) achieves an extremum

\[
\frac{df}{dx} \bigg|_{x_0} = 0
\]

• Two-variables function

\( f(x, y) \) is differentiable in its domain. At \((x_0, y_0), f(x, y)\) achieves an extremum

\[
\frac{\partial f}{\partial x} \bigg|_{(x_0, y_0)} = 0, \quad \frac{\partial f}{\partial y} \bigg|_{(x_0, y_0)} = 0
\]
Lagrange multiplier

• In general case

If $f(x), x \in \mathbb{R}^{n \times 1}$ achieves a local extremum at $x_0$ and it is derivable at $x_0$, then $x_0$ is a stationary point of $f(x)$, i.e.,

$$\frac{\partial f}{\partial x_1} |_{x_0} = 0, \frac{\partial f}{\partial x_2} |_{x_0} = 0, \ldots, \frac{\partial f}{\partial x_n} |_{x_0} = 0$$

Or in other words,

$$\nabla f(x) |_{x=x_0} = 0$$
Lagrange multiplier

- Lagrange multiplier is a strategy for finding the stationary point of a function subject to equality constraints

Problem: find stationary points for \( y = f(x), \, x \in \mathbb{R}^{n \times 1} \)
under \( m \) constraints \( g_k(x) = 0, \, k = 1, 2, \ldots, m \)

Solution:

\[
F(x; \lambda_1, \ldots, \lambda_m) = f(x) + \sum_{k=1}^{m} \lambda_k g_k(x)
\]

If \( (x_0, \lambda_{10}, \lambda_{20}, \ldots, \lambda_{m0}) \) is a stationary point of \( F \), then,
\( x_0 \) is a stationary point of \( f(x) \) with constraints

Joseph-Louis Lagrange
Jan. 25, 1736~Apr. 10, 1813
Lagrange multiplier

• Lagrange multiplier is a strategy for finding the stationary point of a function subject to equality constraints.

Problem: find stationary points for \( y = f(x), \ x \in \mathbb{R}^{n \times 1} \)
under \( m \) constraints \( g_k(x) = 0, k = 1, 2, \ldots, m \)

Solution:
\[
F(x; \lambda_1, \ldots, \lambda_m) = f(x) + \sum_{k=1}^{m} \lambda_k g_k(x)
\]

\((x_0, \lambda_{10}, \ldots, \lambda_{m0})\) is a stationary point of \( F \)

\[
\frac{\partial F}{\partial x_1} = 0, \quad \frac{\partial F}{\partial x_2} = 0, \ldots, \quad \frac{\partial F}{\partial x_n} = 0, \quad \frac{\partial F}{\partial \lambda_1} = 0, \quad \frac{\partial F}{\partial \lambda_2} = 0, \ldots, \quad \frac{\partial F}{\partial \lambda_m} = 0
\]
at that point

n + m equations!
Lagrange multiplier

• Example

Problem: for a given point $p_0 = (1, 0)$, among all the points lying on the line $y = x$, identify the one having the least distance to $p_0$.

The distance is

$$f(x, y) = (x - 1)^2 + (y - 0)^2$$

Now we want to find the stationary point of $f(x, y)$ under the constraint

$$g(x, y) = y - x = 0$$

According to Lagrange multiplier method, construct another function

$$F(x, y, \lambda) = f(x) + \lambda g(x) = (x - 1)^2 + y^2 + \lambda(y - x)$$

Find the stationary point for $F(x, y, \lambda)$
Lagrange multiplier

• Example

Problem: for a given point \( p_0 = (1, 0) \), among all the points lying on the line \( y = x \), identify the one having the least distance to \( p_0 \).

\[
\begin{align*}
\frac{\partial F}{\partial x} &= 0 \\
\frac{\partial F}{\partial y} &= 0 \\
\frac{\partial F}{\partial \lambda} &= 0
\end{align*}
\]

\[
\begin{align*}
2(x - 1) + \lambda &= 0 \\
2y - \lambda &= 0 \\
x - y &= 0
\end{align*}
\]

\[
\begin{align*}
x &= 0.5 \\
y &= 0.5 \\
\lambda &= 1
\end{align*}
\]

\( (0.5, 0.5, 1) \) is a stationary point of \( F(x, y, \lambda) \)

\( (0.5, 0.5) \) is a stationary point of \( f(x, y) \) under constraints
Content

- Scale Invariant Feature Transform
- Case Study: Homography Estimation
  - Matrix Differentiation
  - Lagrange Multiplier
- Least-squares for Linear Systems
- RANSAC-based Homography Estimation
Consider the following linear equations system

\[
\begin{align*}
    x_1 + x_2 &= 3 \\
    2x_1 + x_2 &= 4
\end{align*}
\]

\[
\begin{bmatrix}
    1 & 1 \\
    2 & 1
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2
\end{bmatrix} =
\begin{bmatrix}
    3 \\
    4
\end{bmatrix}
\]

Matrix form: \( A x = b \)

It can be easily solved

\[
\begin{align*}
    x_1 &= 1 \\
    x_2 &= 2
\end{align*}
\]
LS for Inhomogeneous Linear System

How about the following one?

\[
\begin{align*}
    x_1 + x_2 &= 3 \\
    2x_1 + x_2 &= 4 \\
    x_1 + 2x_2 &= 6
\end{align*}
\]

\[
\begin{bmatrix}
    1 & 1 \\
    2 & 1 \\
    1 & 2
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2
\end{bmatrix}
= \begin{bmatrix}
    3 \\
    4 \\
    6
\end{bmatrix}
\]

It does not have a solution!

What is the condition for a linear equation system \( Ax = b \) can be solved?

Can we solve it in an approximate way?

A: we can use least squares technique!

Carl Friedrich Gauss
LS for Inhomogeneous Linear System

Let's consider a system of $p$ linear equations with $q$ unknowns

\[
\begin{align*}
    a_{11}x_1 + a_{12}x_2 + \ldots + a_{1q}x_q &= b_1 \\
    a_{21}x_1 + a_{22}x_2 + \ldots + a_{2q}x_q &= b_2 \\
    \vdots & \quad \vdots \\
    a_{p1}x_1 + a_{p2}x_2 + \ldots + a_{pq}x_q &= b_p
\end{align*}
\]

\[\iff \quad A\mathbf{x} = \mathbf{b} \]

We consider the case: $p > q$, and $\text{rank}(A)=q$

**In general case, there is no solution!**

Instead, we want to find a vector $\mathbf{x}$ that minimizes the error:

\[
E(\mathbf{x}) = \sum_{i=1}^{p} (a_{i1}x_1 + \ldots + a_{iq}x_q - b_i)^2 = \left|A\mathbf{x} - \mathbf{b}\right|^2
\]
LS for Inhomogeneous Linear System

\[ \mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x}) = \arg\min_{\mathbf{x}} \| \mathbf{A}\mathbf{x} - \mathbf{b} \|_2^2 \]

\[ \mathbf{x}^* = \left( \mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{A}^T \mathbf{b} \]

Pseudoinverse of \( \mathbf{A} \)

*How about the pseudoinverse of \( \mathbf{A} \) when \( \mathbf{A} \) is square and non-singular?*
Let’s consider a system of $p$ linear equations with $q$ unknowns

\[
\begin{align*}
  a_{11}x_1 + a_{12}x_2 + \ldots + a_{1q}x_q &= 0 \\
  a_{21}x_1 + a_{22}x_2 + \ldots + a_{2q}x_q &= 0 \\
  \vdots & \\
  a_{p1}x_1 + a_{p2}x_2 + \ldots + a_{pq}x_q &= 0
\end{align*}
\]

We consider the case: $p>q$, and $\text{rank}(A)=q$

Theoretically, there is only a trivial solution: $x = 0$

So, we add a constraint $\|x\|_2 = 1$ to avoid the trivial solution

\[
\iff A \begin{bmatrix} x \\ \end{bmatrix} = 0
\]

unknowns
LS for Homogeneous Linear System

We want to minimize $E(\mathbf{x}) = \|A\mathbf{x}\|^2$, subject to $\|\mathbf{x}\|^2 = 1$

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} E(\mathbf{x}), \text{ s.t., } \|\mathbf{x}\|^2 = 1 \quad (1)$$

Use the Lagrange multiplier to solve it,

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \left[ \|A\mathbf{x}\|^2 + \lambda \left(1 - \|\mathbf{x}\|^2\right)\right] \quad (2)$$

Solving the stationary point of the Lagrange function,

$$\begin{cases} 
\frac{\partial}{\partial \mathbf{x}} \left[ \|A\mathbf{x}\|^2 + \lambda \left(1 - \|\mathbf{x}\|^2\right)\right] = 0 \\
\frac{\partial}{\partial \lambda} \left[ \|A\mathbf{x}\|^2 + \lambda \left(1 - \|\mathbf{x}\|^2\right)\right] = 0 
\end{cases} \quad (3)$$
LS for Homogeneous Linear System

\[
\frac{\partial}{\partial x} \left[ \|Ax\|^2_2 + \lambda \left( 1 - \|x\|^2_2 \right) \right] = 0 \tag{3}
\]

Then, we have

\[
A^T A x = \lambda x
\]

\(x\) is the eigen-vector of \(A^T A\) associated with the eigenvalue \(\lambda\)

\[
E(x) = \|Ax\|^2_2 = x^T A^T A x = x^T \lambda x = \lambda
\]

The unit vector \(x\) is the eigenvector associated with the minimum eigenvalue of \(A^T A\)
Content

- Scale Invariant Feature Transform
- Case Study: Homography Estimation
  - Matrix Differentiation
  - Lagrange Multiplier
  - Least-squares for Linear Systems
- RANSAC-based Homography Estimation
RANSAC-based Homography Estimation

Problem definition:

On two projective planes $P_1$ and $P_2$, there is a set of corresponding points $\{x_i, x'_i\}_{i=1}^n$, and we suppose that there is a homography matrix linking the two planes,

$$x'_i = Hx_i, i = 1, 2, \ldots, n$$

Coordinates of $\{x_i\}$ and $\{x'_i\}$ are known, we need to find $H$

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

Note: $H$ is defined up to a scale factor. In other words, it has 8 DOFs
Note: Theoretically speaking, homography can only be estimated between two planes, i.e., when you use such a technique to stitch two images, image contents should be roughly on the same plane.
4 point-correspondence pairs can uniquely determine a homography matrix since each correspondence pair solves two degrees of freedom.

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

\[
\begin{align*}
    a_{11}x + a_{12}y + a_{13} &= cu \\
    a_{21}x + a_{22}y + a_{23} &= cv \\
    a_{31}x + a_{32}y + a_{33} &= c
\end{align*}
\]

Lin ZHANG, SSE, 2021
RANSAC-based Homography Estimation

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

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

Thus, four correspondence pairs generate 8 equations

Lin ZHANG, SSE, 2021
RANSAC-based Homography Estimation

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

\[ Ax = 0 \quad (1) \]

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

Line fitting: least square
RANSAC-based Homography Estimation

Line fitting: RANSAC
RANSAC-based Homography Estimation

Line fitting by RANSAC
RANSAC-based Homography Estimation

- Line fitting by RANSAC
  - Randomly select two points
RANSAC-based Homography Estimation

Line fitting by RANSAC

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

Line fitting by RANSAC

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

Line fitting by RANSAC

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

Line fitting by RANSAC

• Test another two points
RANSAC-based Homography Estimation

Line fitting by RANSAC

The final fitting result
RANSAC-based Homography Estimation

Line fitting by RANSAC

- The final fitting result
Can you describe the steps of homography estimation when using RANSAC?
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.
Transform image one using the estimated homography matrix
Homography Estimation: Example 1

Finally, stitch the transformed image one with image two
Homography Estimation: Example 2
Homography Estimation: Example 2

Interest points detection
Correspondence estimation

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

Lin ZHANG, SSE, 2021