Lecture 3
Local Feature Descriptors and Matching

Lin ZHANG
School of Software Engineering
Tongji University
Spring 2022

Content

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

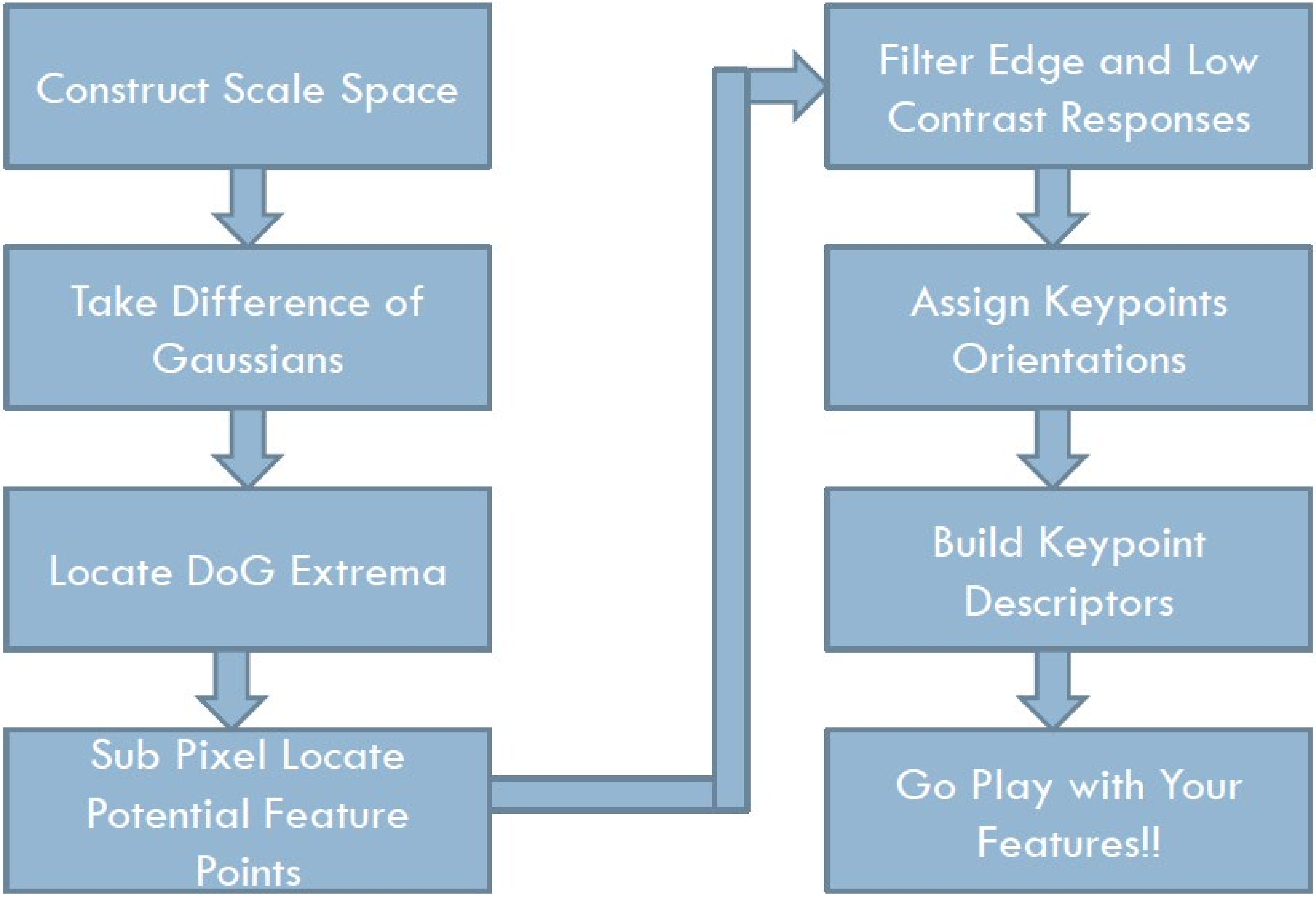
SIFT

- 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

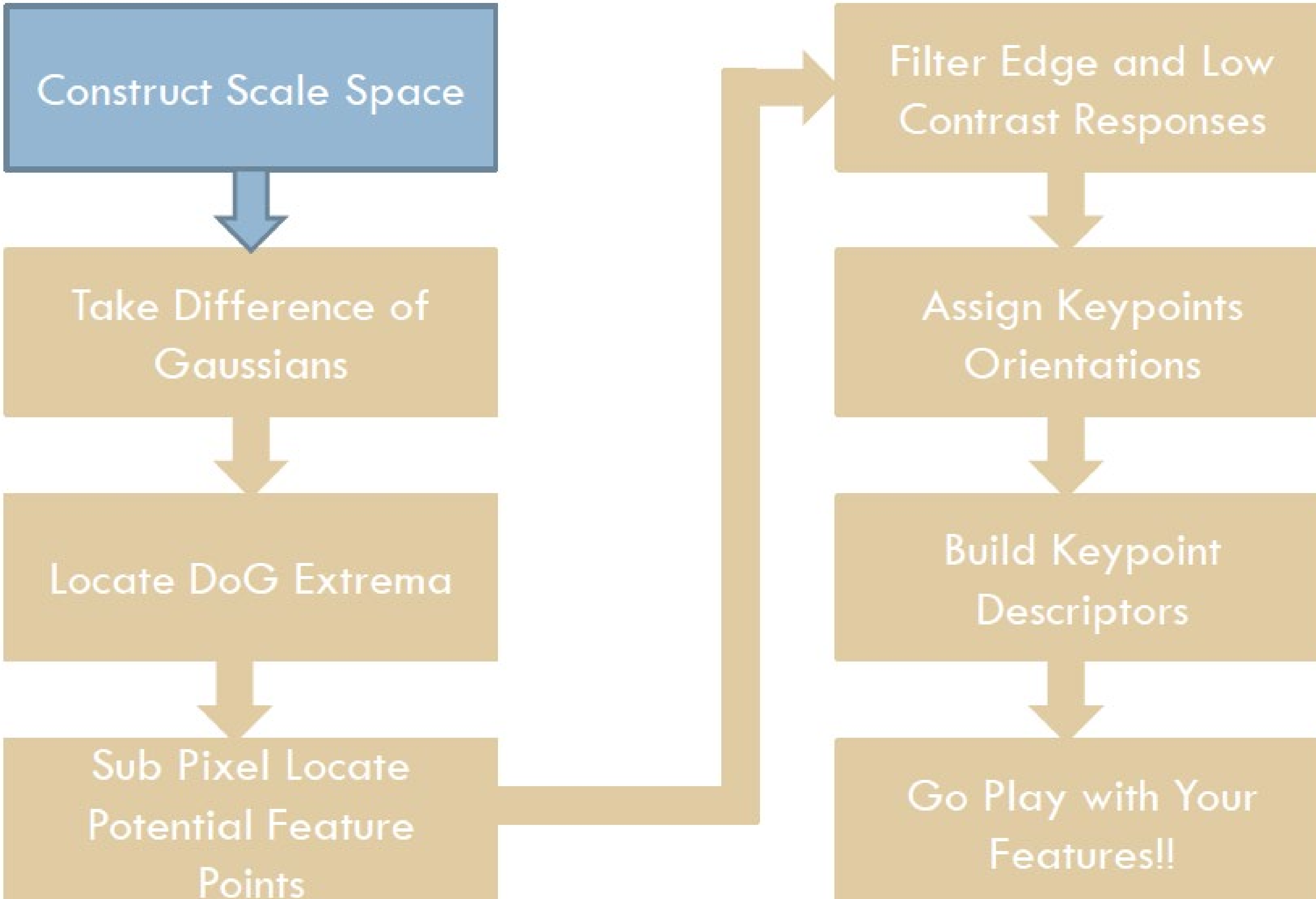
[1] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60 (2), pp. 91-110, 2004 (cited number: 66075 by Mar. 13, 2022)

Prof. David Lowe
University of British Columbia

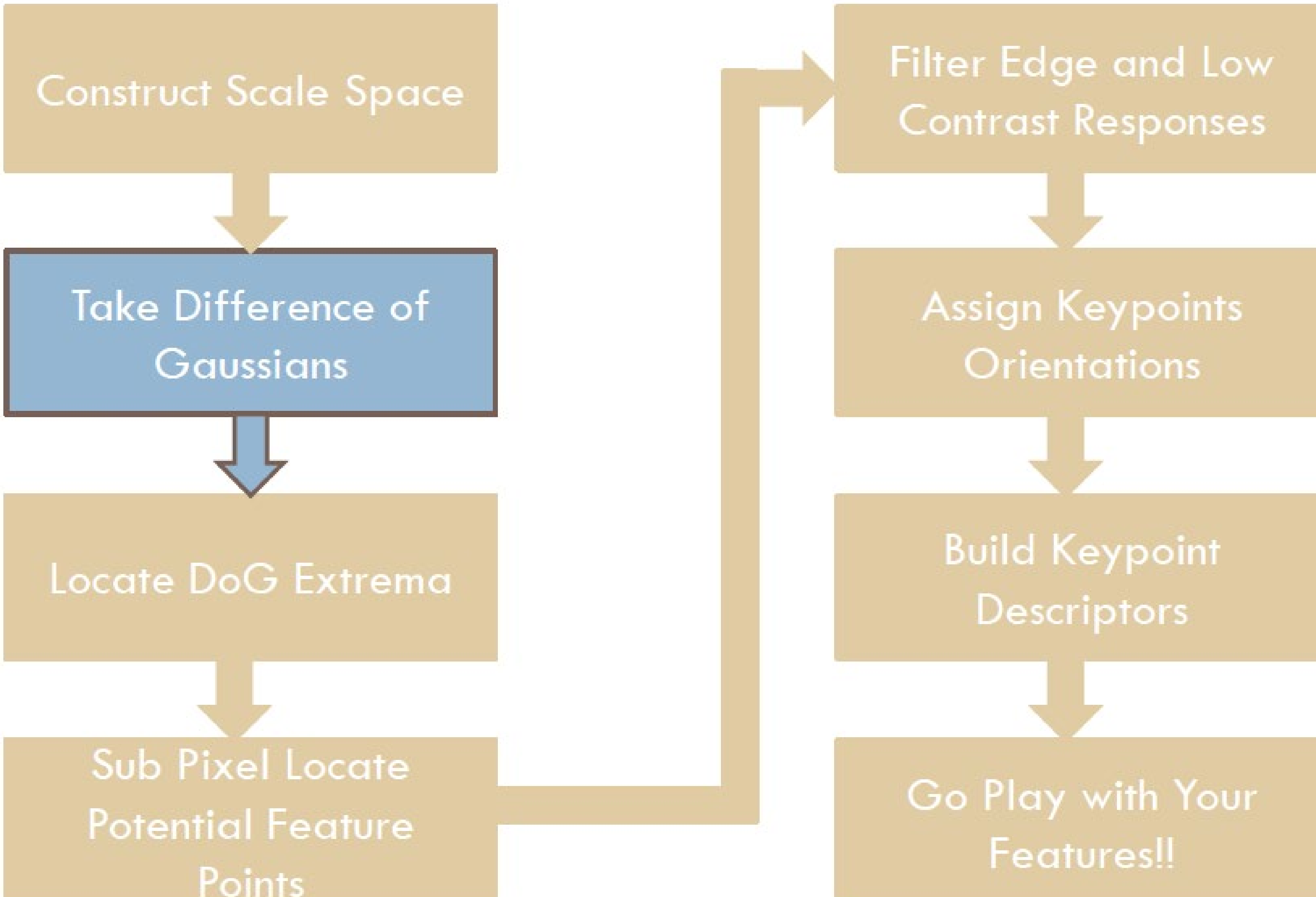
SIFT


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!!

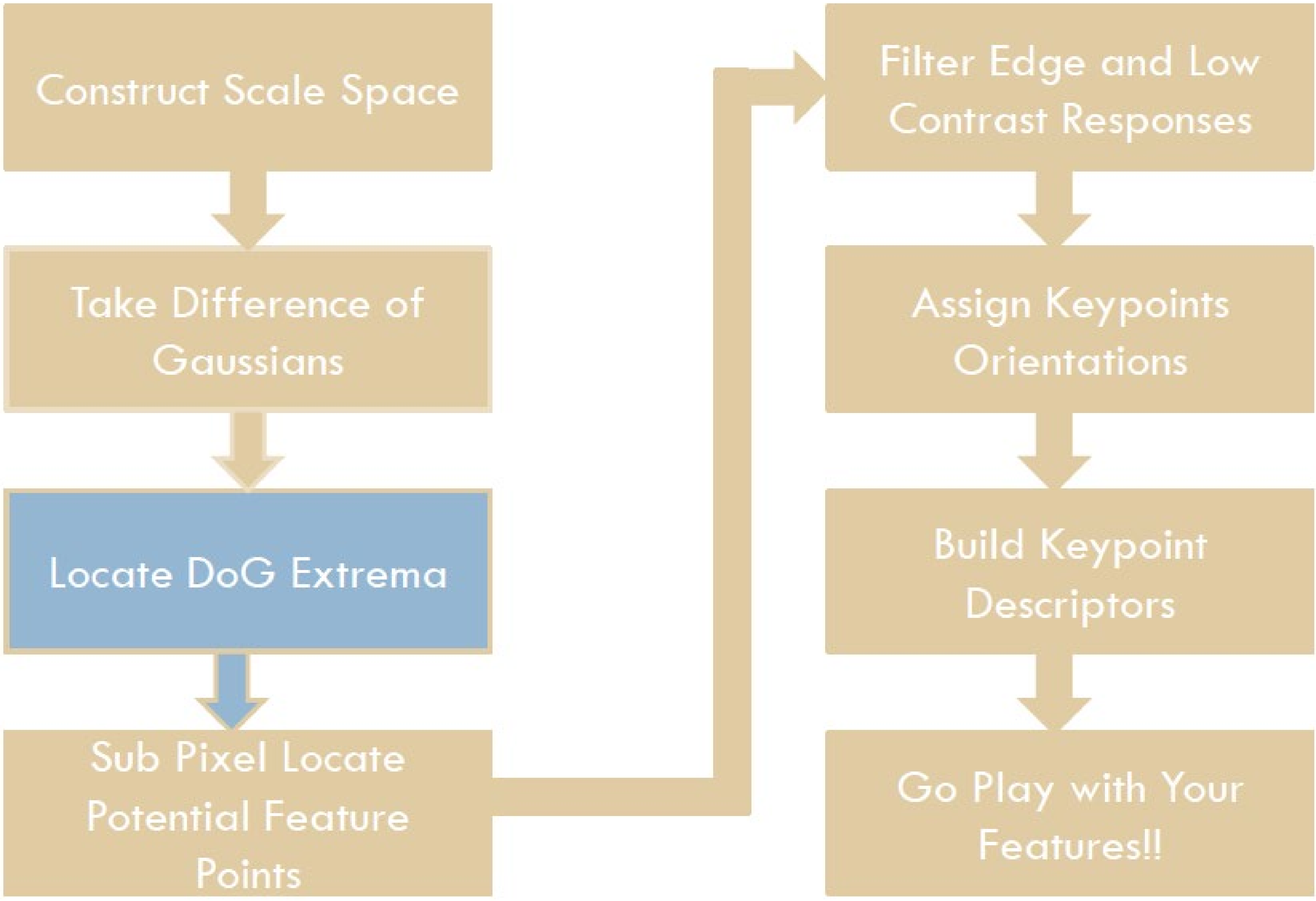
SIFT


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

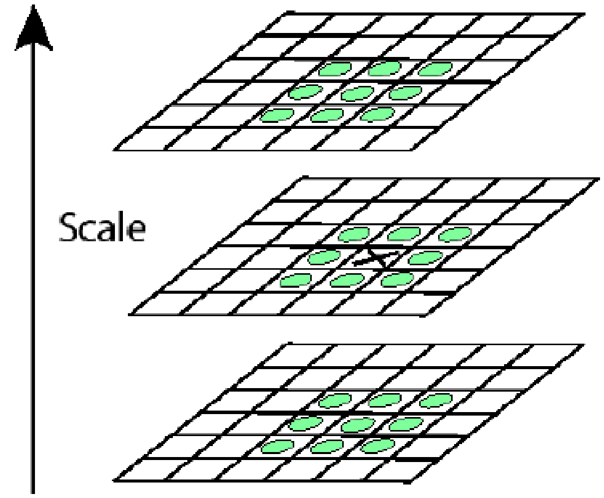

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

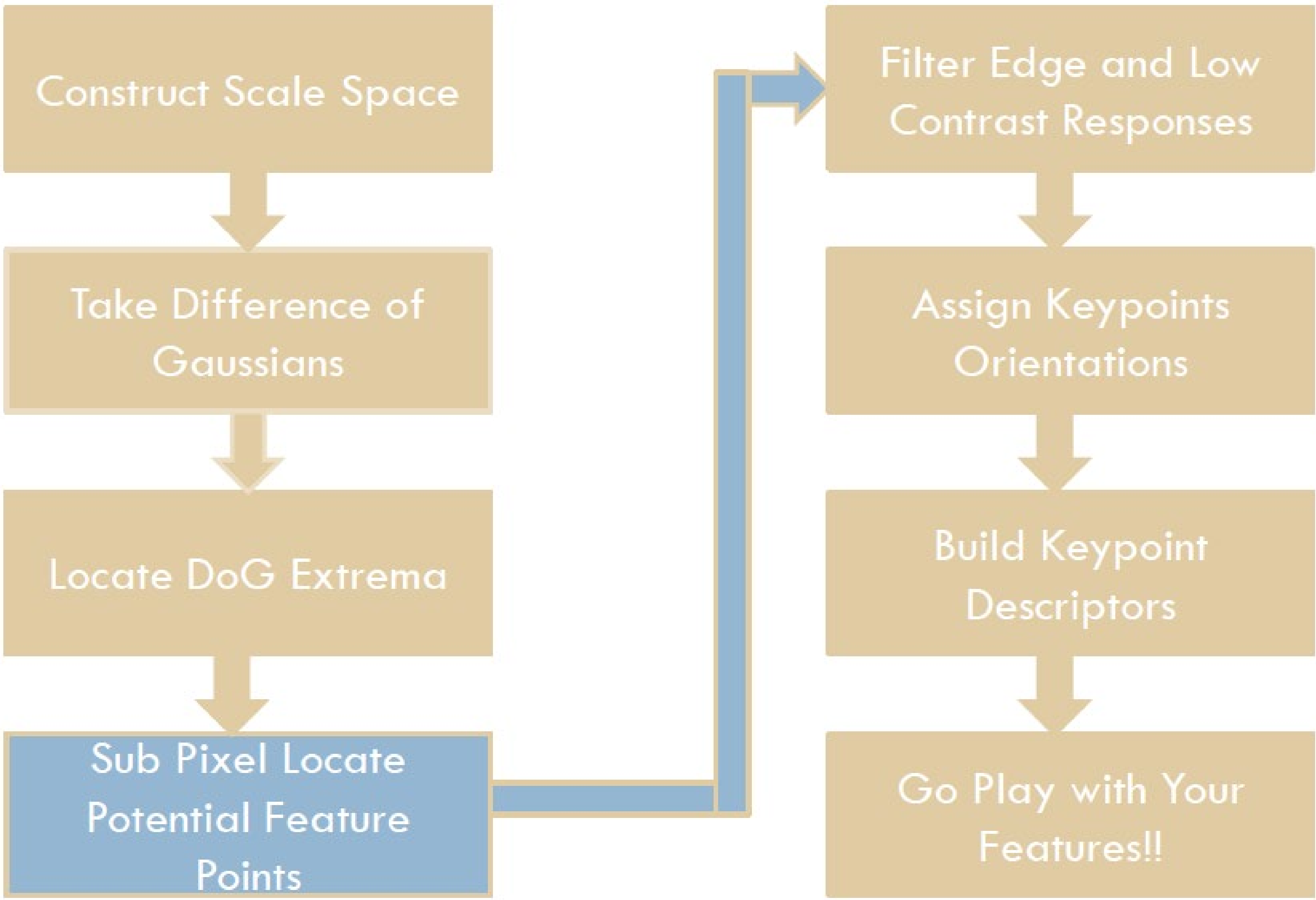

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

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

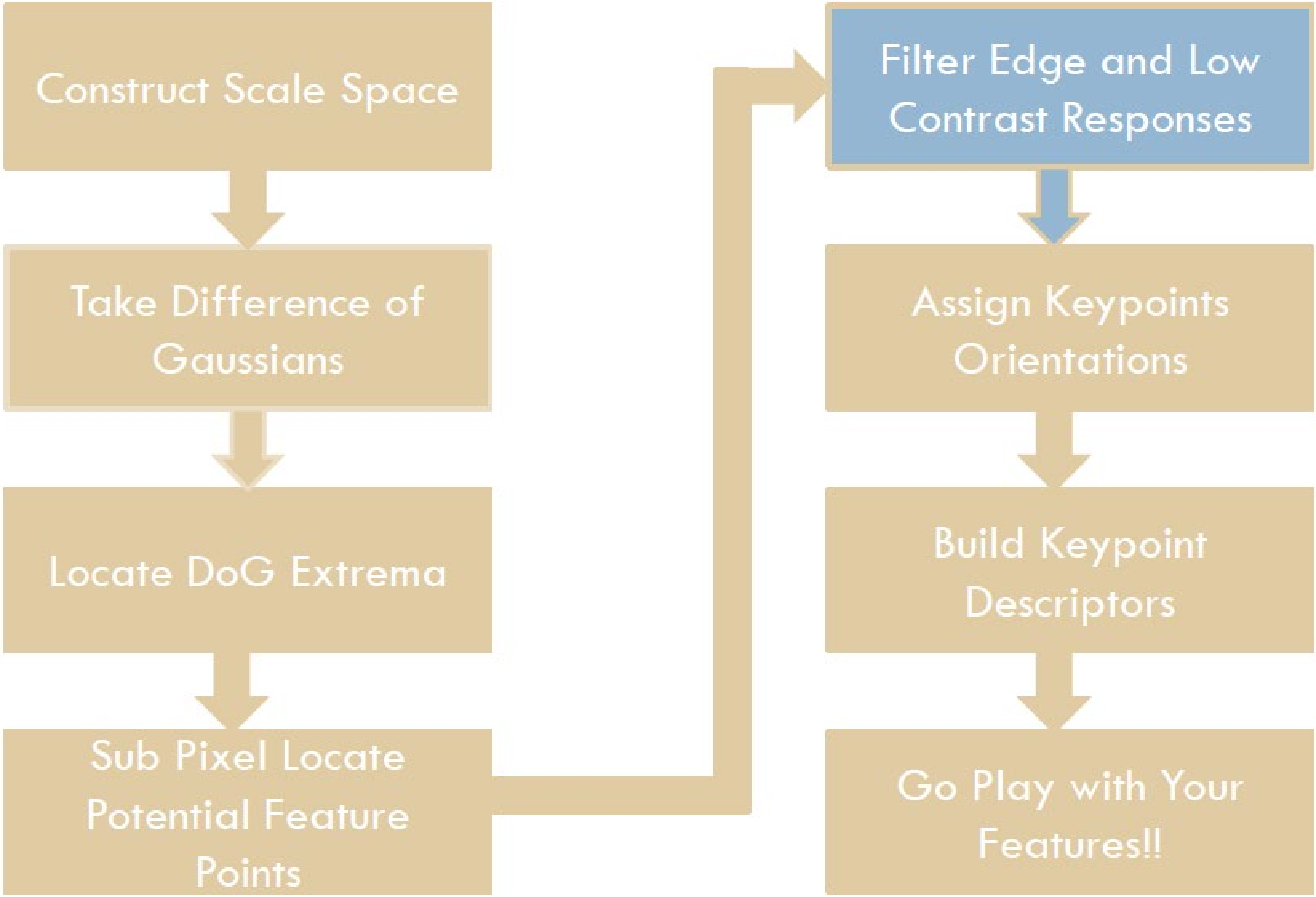


SIFT

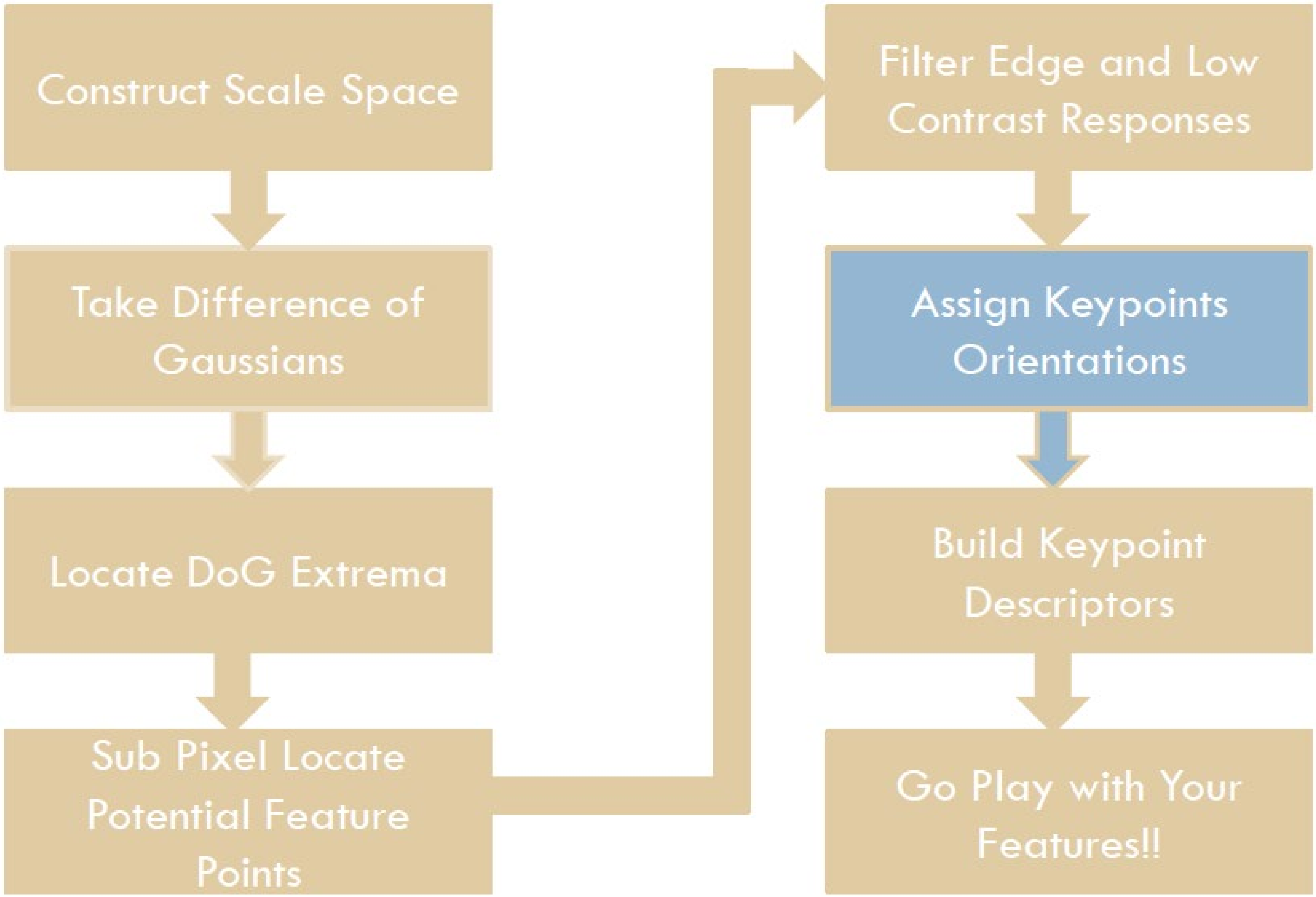


Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

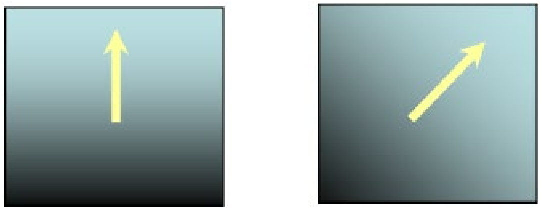

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

SIFT

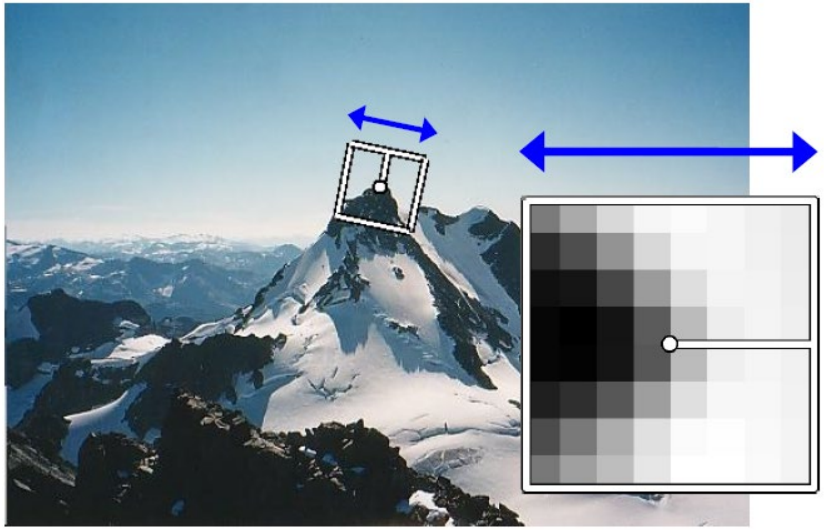

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

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)

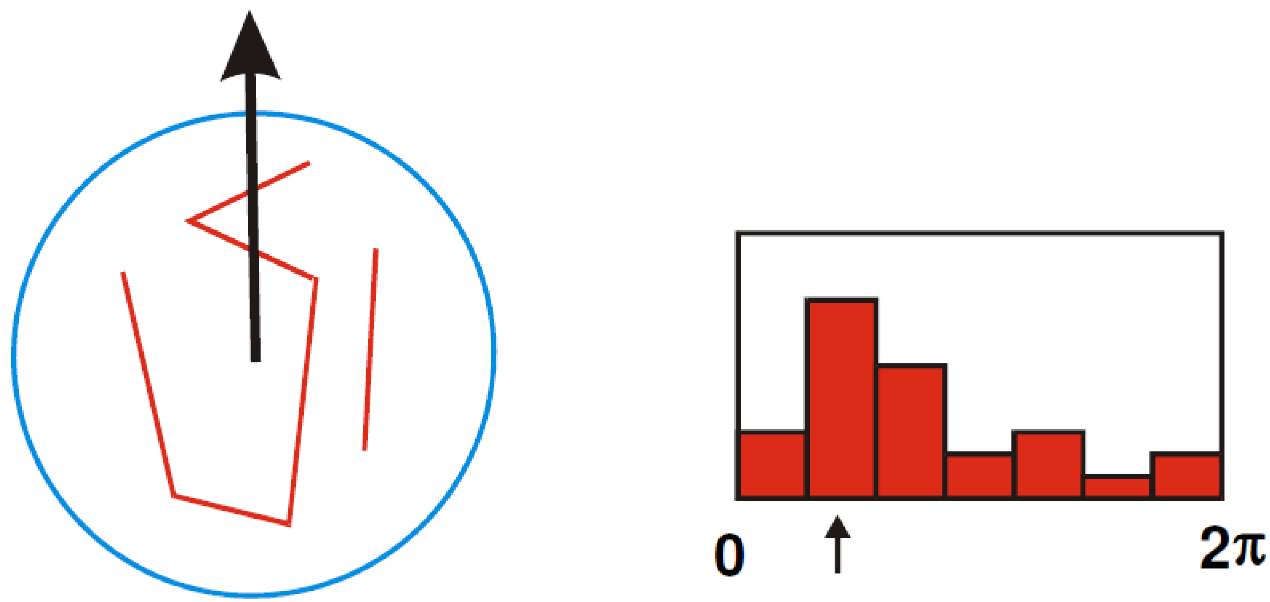


  - Rotate the patch according to this angle; this can achieve rotation invariance description

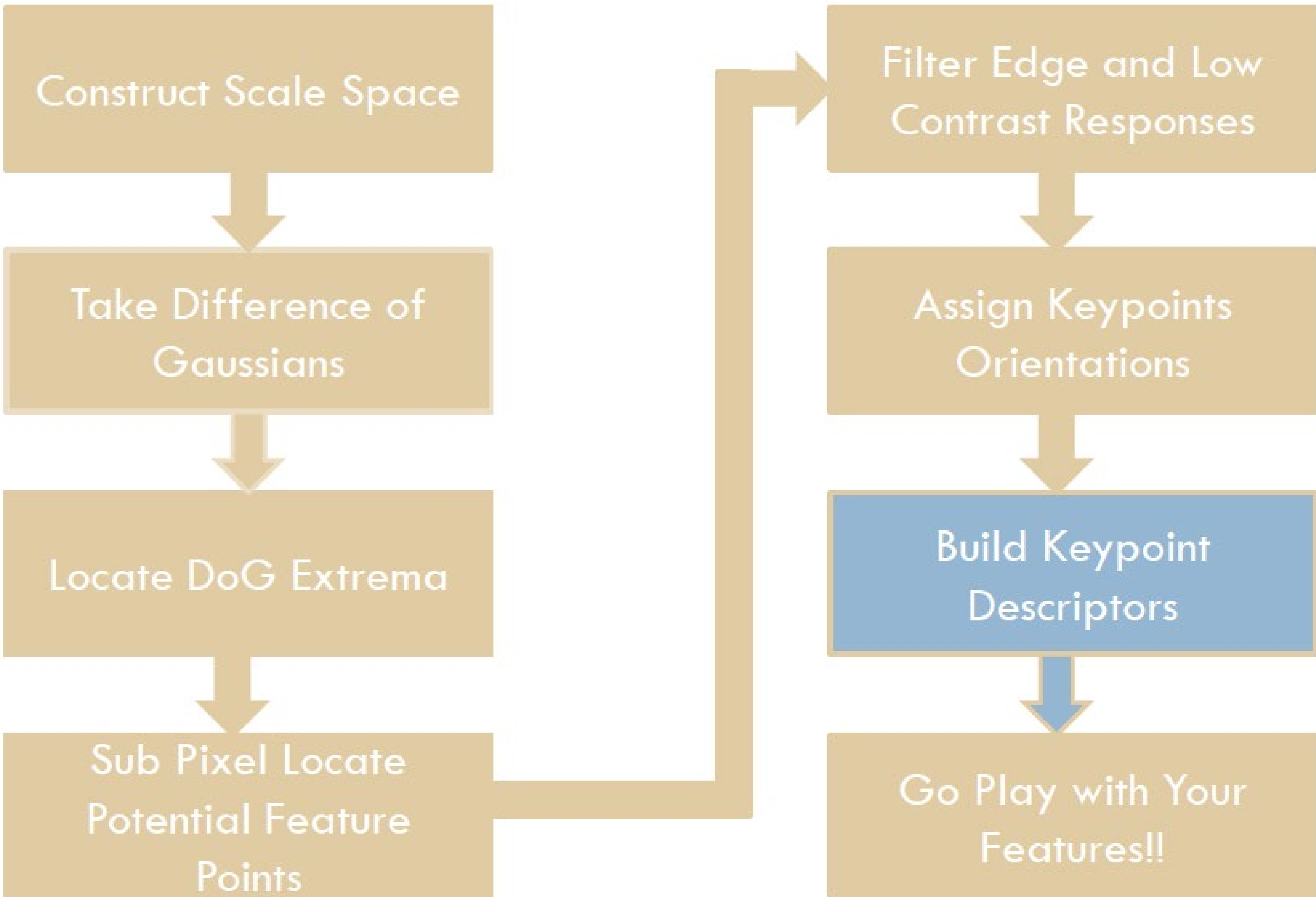


Assign Keypoints Orientations

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



SIFT

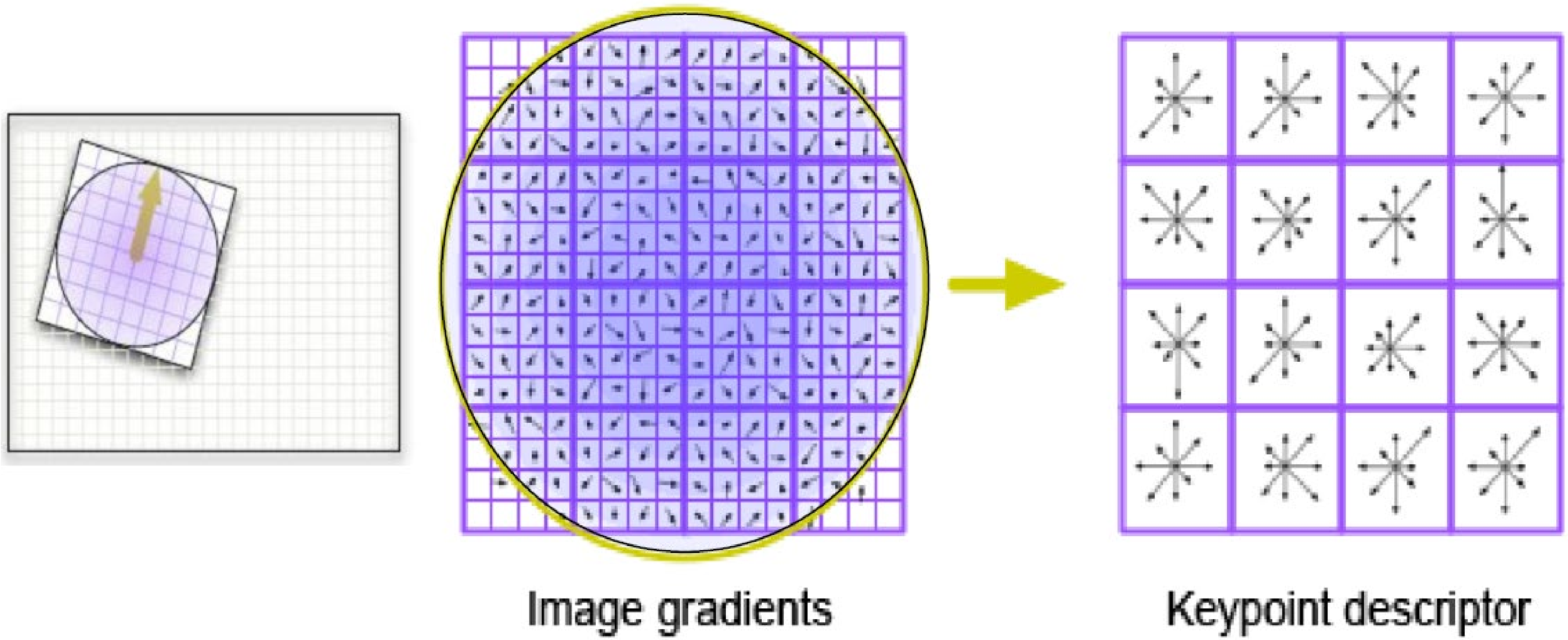

Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

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

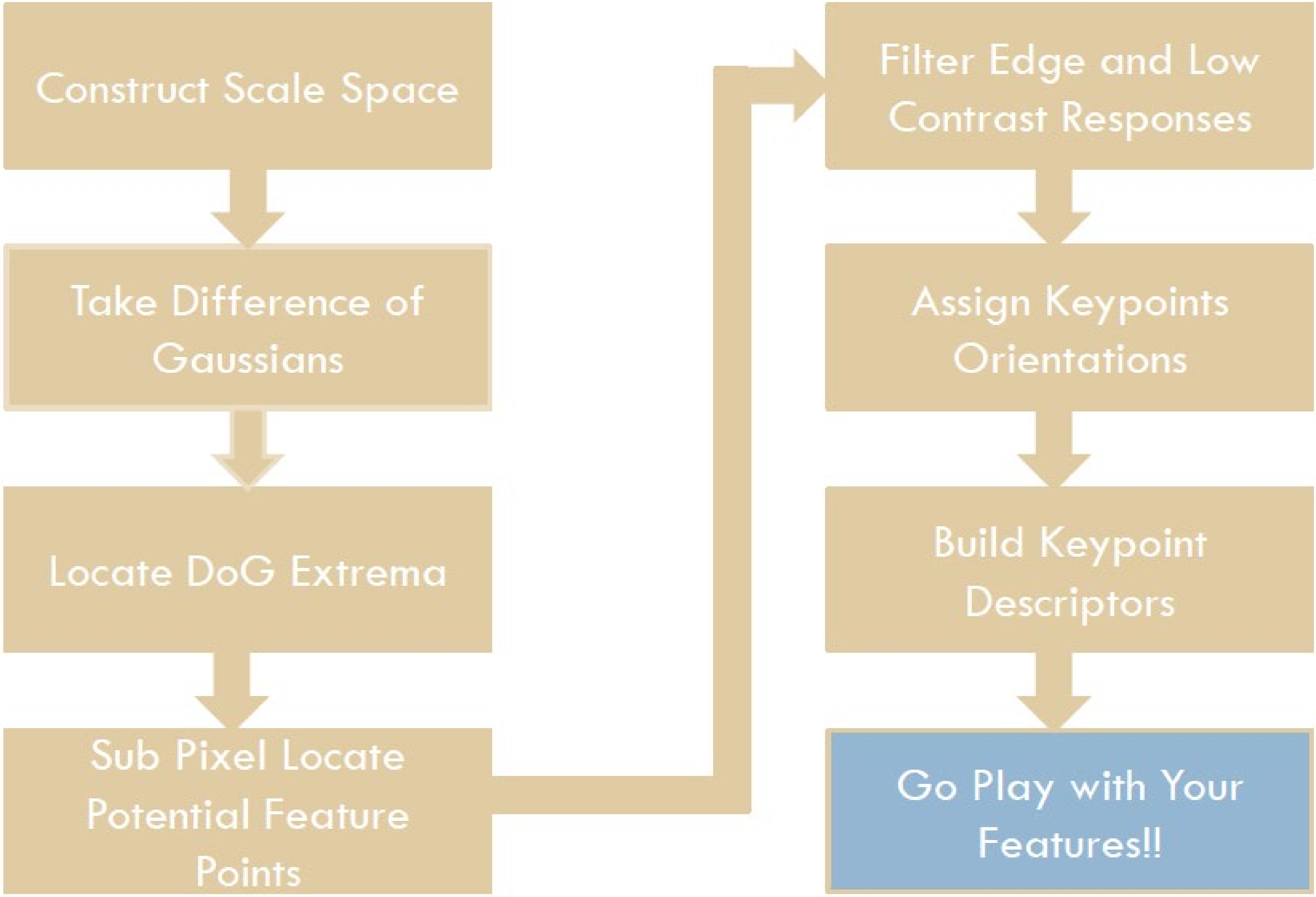


- Actual implementation uses 4*4 sub regions which lead to a 4*4*8 = 128 element vector

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]

SIFT



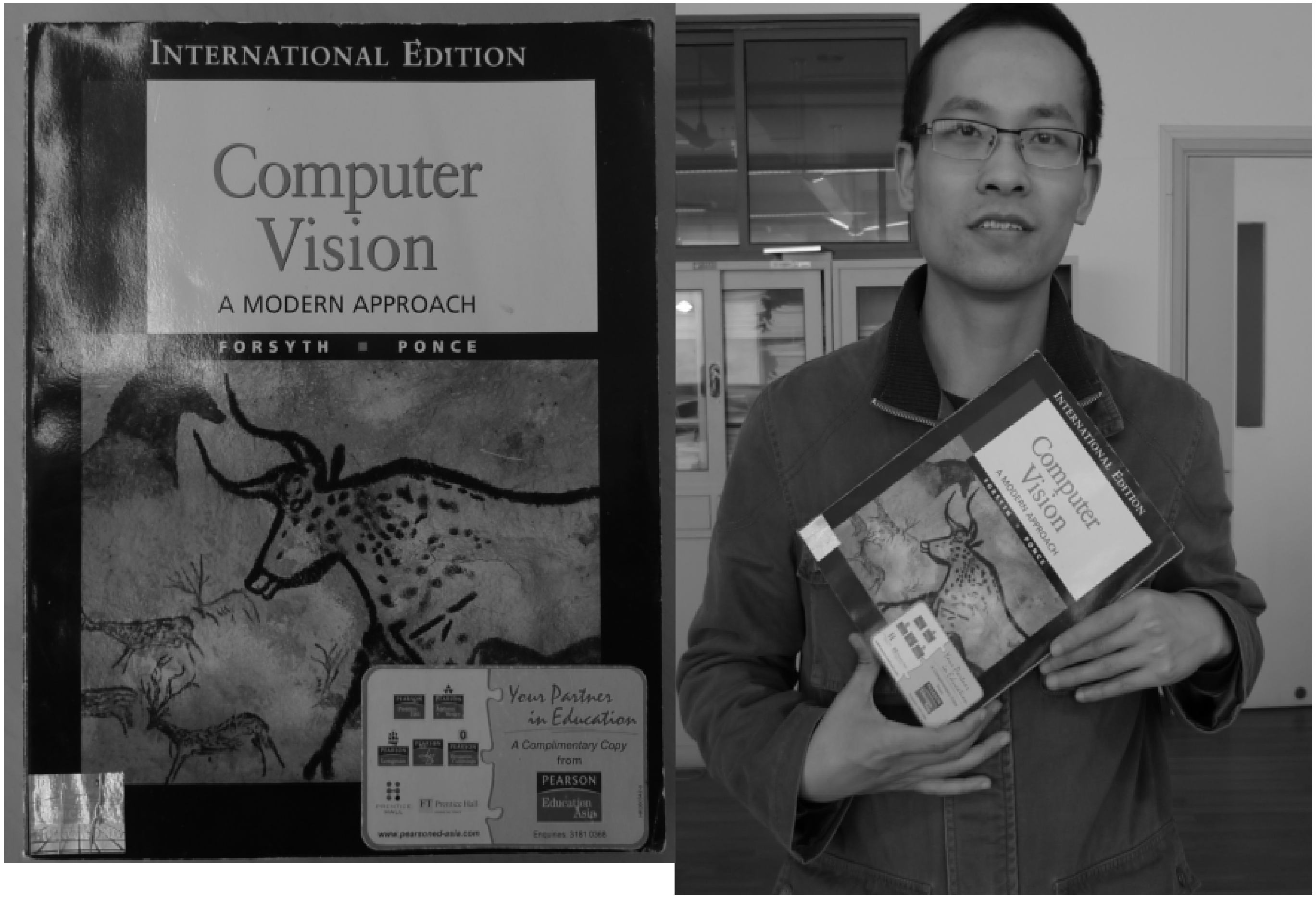
Construct Scale Space
Take Difference of Gaussians
Locate DoG Extrema
Sub Pixel Locate Potential Feature Points
Filter Edge and Low Contrast Responses
Assign Keypoints Orientations
Build Keypoint Descriptors
Go Play with Your Features!!

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

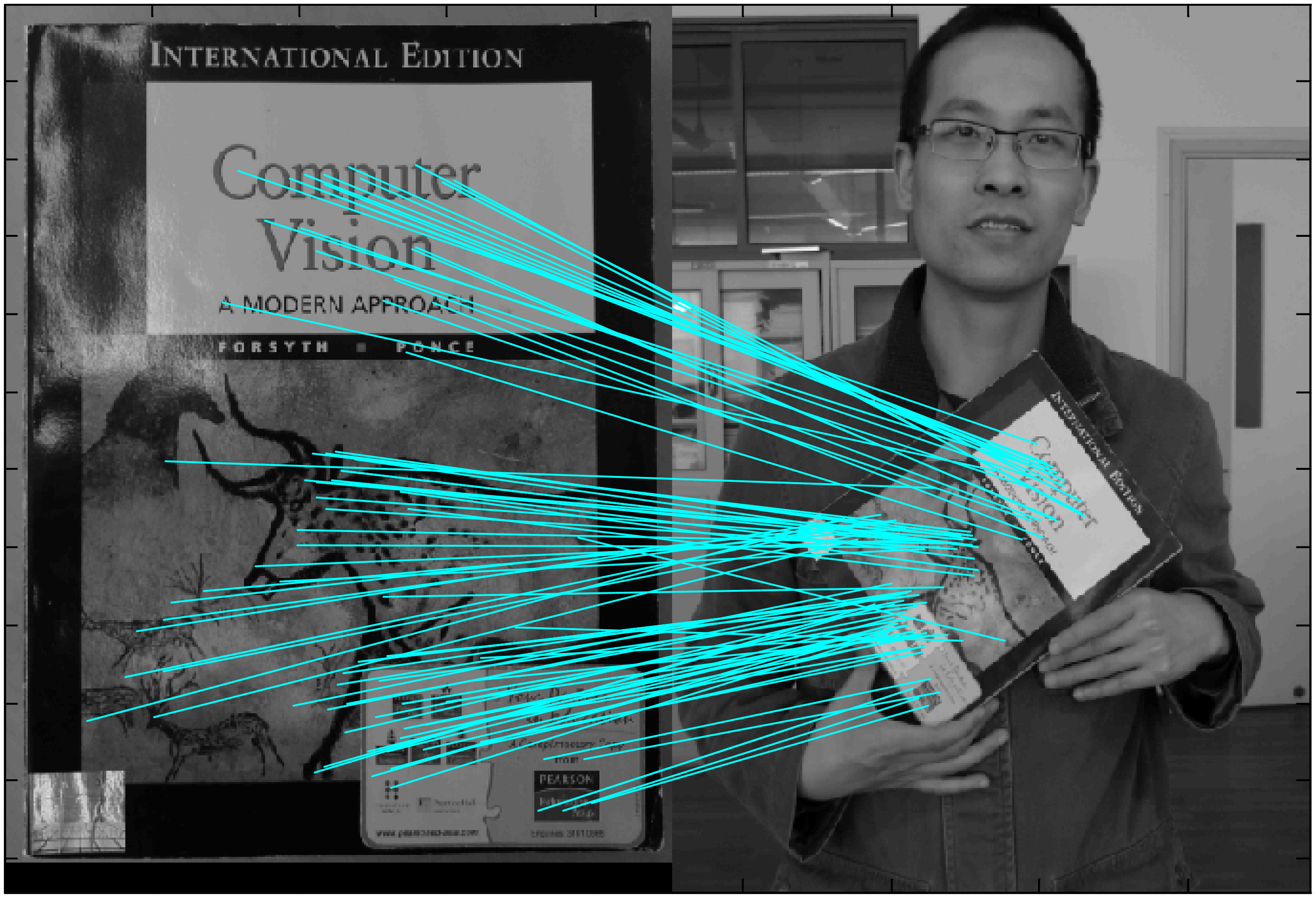
Applications of SIFT

- Object recognition



Applications of SIFT

- Object recognition



Content

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

Matrix differentiation

- Function is a vector and the variable is a scalar

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

Definition

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

Matrix differentiation

- Function is a matrix and the variable is a scalar

$$\mathbf{F}(t)=\begin{bmatrix} f_{11}(t)\ f_{12}(t),\ldots,f_{1m}(t) \\ f_{21}(t)\ f_{22}(t),\ldots,f_{2m}(t) \\ \vdots \\ f_{n1}(t)\ f_{n2}(t),\ldots,f_{nm}(t) \end{bmatrix}=\left[f_{ij}(t)\right]_{n\times m}$$

Definition

$$\frac{d\mathbf{F}}{dt}=\begin{bmatrix} \frac{df_{11}(t)}{dt}\ \frac{df_{12}(t)}{dt},\ldots,\frac{df_{1m}(t)}{dt} \\ \frac{df_{21}(t)}{dt}\ \frac{df_{22}(t)}{dt},\ldots,\frac{df_{2m}(t)}{dt} \\ \vdots \\ \frac{df_{n1}(t)}{dt}\ \frac{df_{n2}(t)}{dt},\ldots,\frac{df_{nm}(t)}{dt} \end{bmatrix}=\left[\frac{df_{ij}(t)}{dt}\right]_{n\times m}$$

Matrix differentiation

- Function is a scalar and the variable is a vector

$$f(\mathbf{x}), \mathbf{x} = (x_1, x_2, ..., x_n)^T$$

Definition

$$\frac{df}{d\mathbf{x}} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n} \right]^T$$

In a similar way,

$$f(\mathbf{x}), \mathbf{x} = (x_1, x_2, ..., x_n)$$

$$\frac{df}{d\mathbf{x}} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n} \right]$$

Matrix differentiation

- Function is a vector and the variable is a vector

$$\mathbf{x} = \left[x_1, x_2, \ldots, x_n\right]^T, \mathbf{y} = \left[y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_m(\mathbf{x})\right]^T$$

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

$$\mathbf{x} = \left[x_1, x_2, \ldots, x_n \right]^T, \mathbf{y} = \left[y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_m(\mathbf{x}) \right]^T$$

In a similar way,

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

Matrix differentiation

- Function is a vector and the variable is a vector

Example:

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

$$\frac{d\mathbf{y}^T}{d\mathbf{x}}=\begin{bmatrix} \frac{\partial y_1(\mathbf{x})}{\partial x_1} & \frac{\partial y_2(\mathbf{x})}{\partial x_1} \\ \frac{\partial y_1(\mathbf{x})}{\partial x_2} & \frac{\partial y_2(\mathbf{x})}{\partial x_2} \\ \frac{\partial y_1(\mathbf{x})}{\partial x_3} & \frac{\partial y_2(\mathbf{x})}{\partial x_3} \end{bmatrix}=\begin{bmatrix} 2x_1 & 0 \\ -1 & 3 \\ 0 & 2x_3 \end{bmatrix}$$

Matrix differentiation

- Function is a scalar and the variable is a matrix

$$f(\mathbf{X}), \mathbf{X} \in \mathbb{R}^{m \times n}$$

Definition

$$\frac{df(\mathbf{X})}{d\mathbf{X}} = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} \frac{\partial f}{\partial x_{12}} \cdots \frac{\partial f}{\partial x_{1n}} \\ \cdots \\ \frac{\partial f}{\partial x_{m1}} \frac{\partial f}{\partial x_{m2}} \cdots \frac{\partial f}{\partial x_{mn}} \end{bmatrix}$$

Matrix differentiation

- Useful results

(1)
$$\mathbf{x}, \mathbf{a} \in \mathbb{R}^{n\times 1}$$

Then,

$$\frac{d\mathbf{a}^T\mathbf{x}}{d\mathbf{x}} = \mathbf{a}, \frac{d\mathbf{x}^T\mathbf{a}}{d\mathbf{x}} = \mathbf{a}$$

How to prove?

Matrix differentiation

- Useful results

(2) $\mathbf{x} \in \mathbb{R}^{n\times 1}$ Then, $\dfrac{d\mathbf{x}^T\mathbf{x}}{d\mathbf{x}} = 2\mathbf{x}$

(3) $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^{m\times 1}$, $\mathbf{x} \in \mathbb{R}^{n\times 1}$, $\dfrac{d\mathbf{y}^T(\mathbf{x})}{d\mathbf{x}} = \left(\dfrac{d\mathbf{y}(\mathbf{x})}{d\mathbf{x}^T}\right)^T$

(4) $A \in \mathbb{R}^{m\times n}, \mathbf{x} \in \mathbb{R}^{n\times 1}$ Then, $\dfrac{dA\mathbf{x}}{d\mathbf{x}^T} = A$

(5) $A \in \mathbb{R}^{m\times n}, \mathbf{x} \in \mathbb{R}^{n\times 1}$ Then, $\dfrac{d\mathbf{x}^T A^T}{d\mathbf{x}} = A^T$

(6) $A \in \mathbb{R}^{n\times n}, \mathbf{x} \in \mathbb{R}^{n\times 1}$ Then, $\dfrac{d\mathbf{x}^T A\mathbf{x}}{d\mathbf{x}} = (A + A^T)\mathbf{x}$

(7) $\mathbf{X} \in \mathbb{R}^{m\times n}, \mathbf{a} \in \mathbb{R}^{m\times 1}, \mathbf{b} \in \mathbb{R}^{n\times 1}$ Then, $\dfrac{d\mathbf{a}^T\mathbf{X}\mathbf{b}}{d\mathbf{X}} = \mathbf{a}\mathbf{b}^T$

Matrix differentiation

- Useful results

(8) $\mathbf{X} \in \mathbb{R}^{n\times m}, \mathbf{a} \in \mathbb{R}^{m\times 1}, \mathbf{b} \in \mathbb{R}^{n\times 1}$ Then, $\frac{d\mathbf{a}^T\mathbf{X}^T\mathbf{b}}{d\mathbf{X}} = \mathbf{b}\mathbf{a}^T$

(9) $\mathbf{X} \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times m}$ Then, $\frac{d\left(tr\mathbf{X}B\right)}{d\mathbf{X}} = B^T$

(10) $\mathbf{X} \in \mathbb{R}^{n\times n}, \mathbf{X}$ is invertible, $\frac{d\left|\mathbf{X}\right|}{d\mathbf{X}} = \left|\mathbf{X}\right|\left(\mathbf{X}^{-1}\right)^T$

Content

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

$$\Rightarrow \frac{df}{dx}\Big|_{x_0} = 0$$

- Two-variables function

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

$$\Rightarrow \frac{\partial f}{\partial x}\Big|_{(x_0, y_0)} = 0, \frac{\partial f}{\partial y}\Big|_{(x_0, y_0)} = 0$$

Lagrange multiplier

- In general case

If $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^{n\times 1}$ achieves a local extremum at $\mathbf{x}_0$ and it is derivable at $\mathbf{x}_0$, then $\mathbf{x}_0$ is a stationary point of $f(\mathbf{x})$, i.g.,

$$\frac{\partial f}{\partial x_1}\bigg|_{\mathbf{x}_0} = 0, \frac{\partial f}{\partial x_2}\bigg|_{\mathbf{x}_0} = 0, \ldots, \frac{\partial f}{\partial x_n}\bigg|_{\mathbf{x}_0} = 0$$

Or in other words,

$$\nabla f(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_0} = \mathbf{0}$$

Lagrange multiplier

- Lagrange multiplier is a strategy for finding **all the possible** extremum points of a function subject to equality constraints

Problem: find **all the possible** extremum points for $y = f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^{n\times 1}$

under m constraints $g_k(\mathbf{x}) = 0, k = 1, 2, ..., m$

Solution: $F(\mathbf{x}; \lambda_1, ..., \lambda_m) = f(\mathbf{x}) + \sum_{k=1}^{m} \lambda_k g_k(\mathbf{x})$

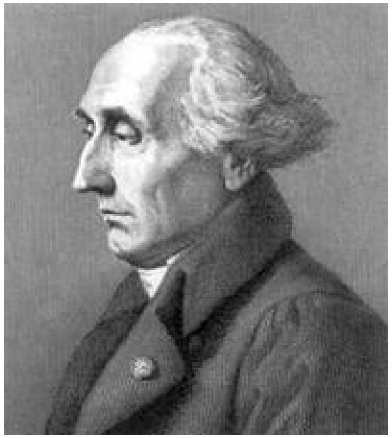
If $\mathbf{x}_0$ is an extremum point of $f(\mathbf{x})$ under constraints

$\Downarrow$

$\exists \lambda_{10}, \lambda_{20} ..., \lambda_{m0}$, making $(\mathbf{x}_0, \lambda_{10}, \lambda_{20} ..., \lambda_{m0})$

a stationary point of F

Thus, by identifying the stationary points of F, we can get all the possible extremum points of $f(\mathbf{x})$ under equality constraints



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

Lagrange multiplier

- Lagrange multiplier is a strategy for finding **all the possible** extremum points of a function subject to equality constraints

Problem: find **all the possible** extremum points for $y = f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^{n\times 1}$

under m constraints $g_k(\mathbf{x}) = 0, k = 1, 2, ..., m$

Solution: $F(\mathbf{x}; \lambda_1, ..., \lambda_m) = f(\mathbf{x}) + \sum_{k=1}^{m} \lambda_k g_k(\mathbf{x})$

$(\mathbf{x}_0, \lambda_{10}, ..., \lambda_{m0})$ is a stationary point of F $\Rightarrow$

$$\underbrace{\frac{\partial F}{\partial x_1} = 0, \frac{\partial F}{\partial x_2} = 0, ..., \frac{\partial F}{\partial x_n} = 0, \frac{\partial F}{\partial \lambda_1} = 0, \frac{\partial F}{\partial \lambda_2} = 0, ..., \frac{\partial F}{\partial \lambda_m} = 0}_{n + m \text{ equations!}}$$

at that point

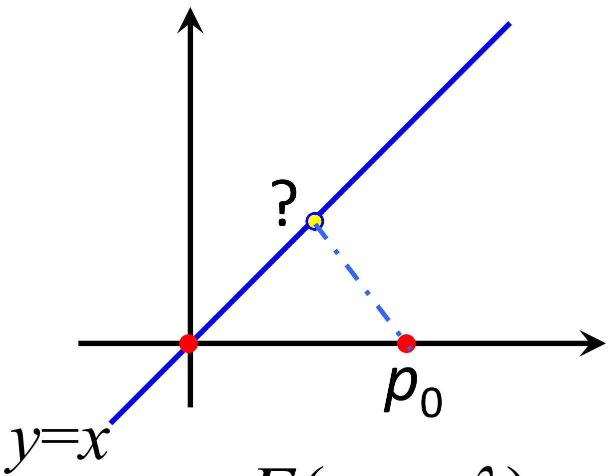
$\Downarrow$

$\mathbf{x}_0$ is a possible extremum point of $f(\mathbf{x})$ under equality constraints

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 global minimizer of $f(x, y)$ under the constraint

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

According to Lagrange multiplier method, construct the Lagrange function

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

Find the stationary point of $F(x, y, \lambda)$

Lagrange multiplier

- Example

$$\begin{cases} \dfrac{\partial F}{\partial x} = 0 \\ \dfrac{\partial F}{\partial y} = 0 \\ \dfrac{\partial F}{\partial \lambda} = 0 \end{cases} \Rightarrow \begin{cases} 2(x-1)+\lambda = 0 \\ 2y - \lambda = 0 \\ x - y = 0 \end{cases} \Rightarrow \begin{cases} x = 0.5 \\ y = 0.5 \\ \lambda = 1 \end{cases}$$

Thus, $(0.5, 0.5, 1)$ is the only stationary point of $F(x, y, \lambda)$

$\Rightarrow$

$(0.5,0.5)$ is the only possible extremum point of $f(x,y)$ under constraints

The global minimizer of $f(x,y)$ under constraints exists

$\Downarrow$

$(0.5,0.5)$ is the global minimizer of $f(x,y)$ under constraints

Content

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

LS for Inhomogeneous Linear System

Consider the following linear equations system

$$\begin{cases} x_1 + x_2 = 3 \\ 2x_1 + x_2 = 4 \end{cases} \Leftrightarrow \underbrace{\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 3 \\ 4 \end{bmatrix}}_{\mathbf{b}}$$

Matrix form: $A\mathbf{x} = \mathbf{b}$

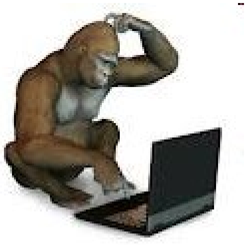
It can be easily solved $\begin{cases} x_1 = 1 \\ x_2 = 2 \end{cases}$

LS for Inhomogeneous Linear System

How about the following one?

$$\begin{cases} x_1 + x_2 = 3 \\ 2x_1 + x_2 = 4 \\ x_1 + 2x_2 = 6 \end{cases} \Leftrightarrow \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix}$$

It does not have a solution!



What is the condition for a linear equation system $A\mathbf{x} = \mathbf{b}$ can be solved?

Carl Friedrich Gauss

Can we solve it in an approximate way?

A: we can use least squares technique!

LS for Inhomogeneous Linear System

Let's consider a system of m linear equations with n unknowns

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = \mathbf{b}_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = \mathbf{b}_2 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = \mathbf{b}_m \end{cases} \Leftrightarrow A_{m\times n}\underset{\text{unknowns}}{\underbrace{\mathbf{x}_{n\times 1}}} = \mathbf{b}_{m\times 1}$$

We consider the case: $m>n$, rank(A)=n, and rank($[A; \mathbf{b}]$)=$n+1$

In general case, there is no solution!

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

$$E(\mathbf{x}) \equiv \sum_{i=1}^{m}(a_{i1}x_1 + \ldots + a_{in}x_n - \mathbf{b}_i)^2 = \|A\mathbf{x} - \mathbf{b}\|_2^2$$

LS for Inhomogeneous Linear System

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

The stationary point of $E(\mathbf{x})$ is $\mathbf{x}_s = \left(A^T A\right)^{-1} A^T \mathbf{b}$

Since $E(\mathbf{x})$ is a **convex** function, its stationary point is the global minimizer[1]

$$\mathbf{x}^* = \mathbf{x}_s = \underbrace{\left(A^T A\right)^{-1} A^T}_{\text{Pseudoinverse of } A} \mathbf{b}$$

Pseudoinverse of A

[1] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004, pp. 69

LS for Homogeneous Linear System

Let's consider a system of m linear equations with n unknowns

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = 0 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = 0 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = 0 \end{cases} \Leftrightarrow A_{m\times n}\boxed{\mathbf{x}}_{n\times 1} = \mathbf{0}$$

unknowns

We consider the case: $m>n$, and rank$(A)=n$

Theoretically, there is only a trivial solution: $\mathbf{x} = 0$

We can add a constraint $\|\mathbf{x}\|_2 = 1$ to avoid the trivial solution

LS for Homogeneous Linear System

We want to minimize $E(\mathbf{x}) = \|A\mathbf{x}\|_2^2$, subject to $\|\mathbf{x}\|_2 = 1$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x}),\ s.t.,\ \|\mathbf{x}\|_2 = 1 \qquad (1)$$

Construct the Lagrange function,

$$L(\mathbf{x}, \lambda) = \|A\mathbf{x}\|_2^2 + \lambda\left(1 - \|\mathbf{x}\|_2^2\right) \qquad (2)$$

Solving the stationary point $(\mathbf{x}_0, \lambda_0)$ of $L(\mathbf{x}, \lambda)$,

$$\begin{cases} \dfrac{\partial\left[\|A\mathbf{x}\|_2^2 + \lambda\left(1 - \|\mathbf{x}\|_2^2\right)\right]}{\partial \mathbf{x}} = \mathbf{0} \\ \dfrac{\partial\left[\|A\mathbf{x}\|_2^2 + \lambda\left(1 - \|\mathbf{x}\|_2^2\right)\right]}{\partial \lambda} = 0 \end{cases} \Rightarrow \begin{cases} A^T A\mathbf{x}_0 = \lambda_0 \mathbf{x}_0 \\ \mathbf{x}_0^T \mathbf{x}_0 = 1 \end{cases}$$

Note: the stationary point of $L(\mathbf{x}, \lambda)$ is not unique

LS for Homogeneous Linear System

Suppose that $(\mathbf{x}_i, \lambda_i)$ is a stationary point of L, then $\mathbf{x}_i$ is a possible extremum point of $E(\mathbf{x})$ under the equality constraint and we have

$$E(\mathbf{x}_i) = \|A\mathbf{x}_i\|_2^2 = \mathbf{x}_i^T A^T A \mathbf{x}_i = \mathbf{x}_i^T \lambda_i \mathbf{x}_i = \lambda_i$$

The global minimum of $E(\mathbf{x})$ is $\min\{\lambda_i\}$ and the global minimizer of $E(\mathbf{x})$ is the unit eigen-vector of $A^T A$ associated with its least eigen value

Content

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

Problem of Homography Estimation

Problem definition:

On two projective planes π_1 and π_2, there is a set of corresponding points $\left\{\mathbf{x}_i, \mathbf{x}_i^{'}\right\}_{i=1}^{n}$, and we suppose that there is a homography matrix linking the two planes,

$$\mathbf{x}_i^{'} = H\mathbf{x}_i, i = 1, 2, ..., n$$

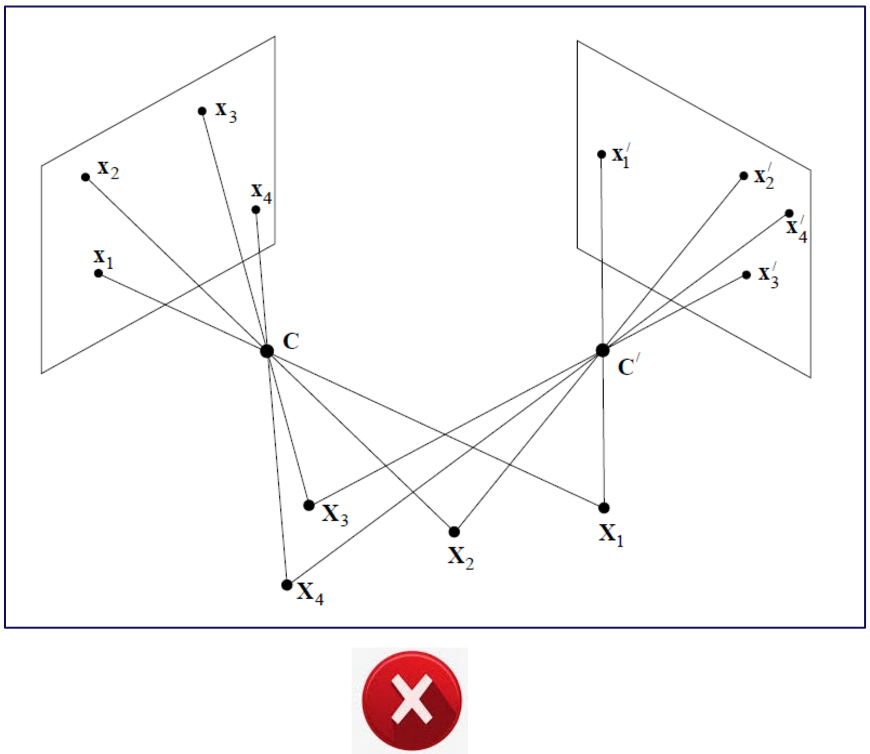
Coordinates of $\left\{\mathbf{x}_i\right\}_i^n$ and $\left\{\mathbf{x}_i^{'}\right\}_{i=1}^{n}$ are known, we need to find H

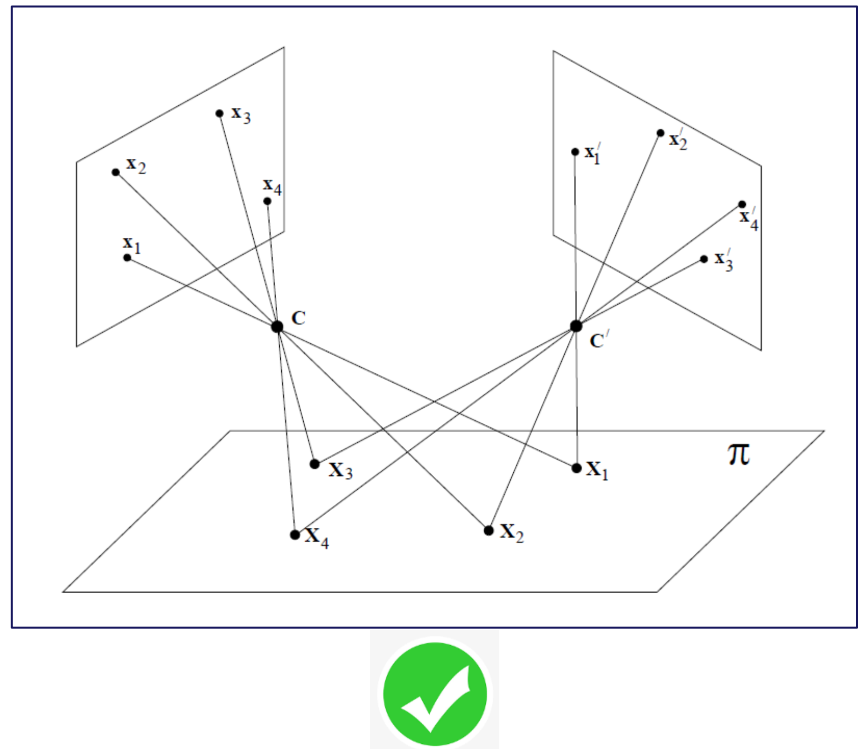
$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Note: H is defined up to a scale factor. In other words, it has 8 DOFs

Problem of Homography Estimation

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





Problem of Homography Estimation

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

$$c\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \Rightarrow \begin{cases} h_{11}u + h_{12}v + h_{13} = cx \\ h_{21}u + h_{22}v + h_{23} = cy \\ h_{31}u + h_{32}v + h_{33} = c \end{cases}$$

$$\Rightarrow \begin{cases} \dfrac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}} = x \\ \dfrac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}} = y \end{cases}$$

Note: here we assume that the matching points are all finite points (no points at infinity)

Problem of Homography Estimation

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

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

Thus, four correspondence pairs generate 8 equations

Problem of Homography Estimation

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

$$\underset{8\times 9}{A}\,\underset{9\times 1}{\mathbf{x}} = 0 \qquad (1)$$

Normally, $Rank(A) = 8$; thus (1) has 1 (9-8) solution vector (nonzero) in its solution space

In our case, since we have n>4 point pairs, we get

$$\mathbf{A}_{2n\times 9}\mathbf{h}_{9\times 1} = \mathbf{0}$$

It is an overdetermined homogeneous linear equation system

Problem of Homography Estimation

Since only the ratios among the elements of H take effect, in another way we can fix h_{33}=1,

$$c\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \Rightarrow \begin{cases} h_{11}u + h_{12}v + h_{13} = cx \\ h_{21}u + h_{22}v + h_{23} = cy \\ h_{31}u + h_{32}v + 1 = c \end{cases} \Rightarrow \begin{cases} \dfrac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + 1} = x \\ \dfrac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + 1} = y \end{cases}$$

$$\Rightarrow \begin{pmatrix} u & v & 1 & 0 & 0 & 0 & -ux & -vx \\ 0 & 0 & 0 & u & v & 1 & -uy & -vy \end{pmatrix}\begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$\Rightarrow$ Since we have n>4 point pairs, we get

$$\mathbf{A}_{2n\times 8}\mathbf{h}_{8\times 1} = \mathbf{b}_{2n\times 1}$$

It is an overdetermined inhomogeneous linear equation system

Content

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

RANSAC-based Homography Estimation

- When there are more than 4 correspondence pairs, is it a proper way to use the LS method to solve the model directly?
  - NO! Because usually, outliers exist among the correspondence pairs

RANdom SAmple Consensus (RANSAC) is an iterative framework to estimate a parametric model from observations with noisy outliers

RANSAC-based 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

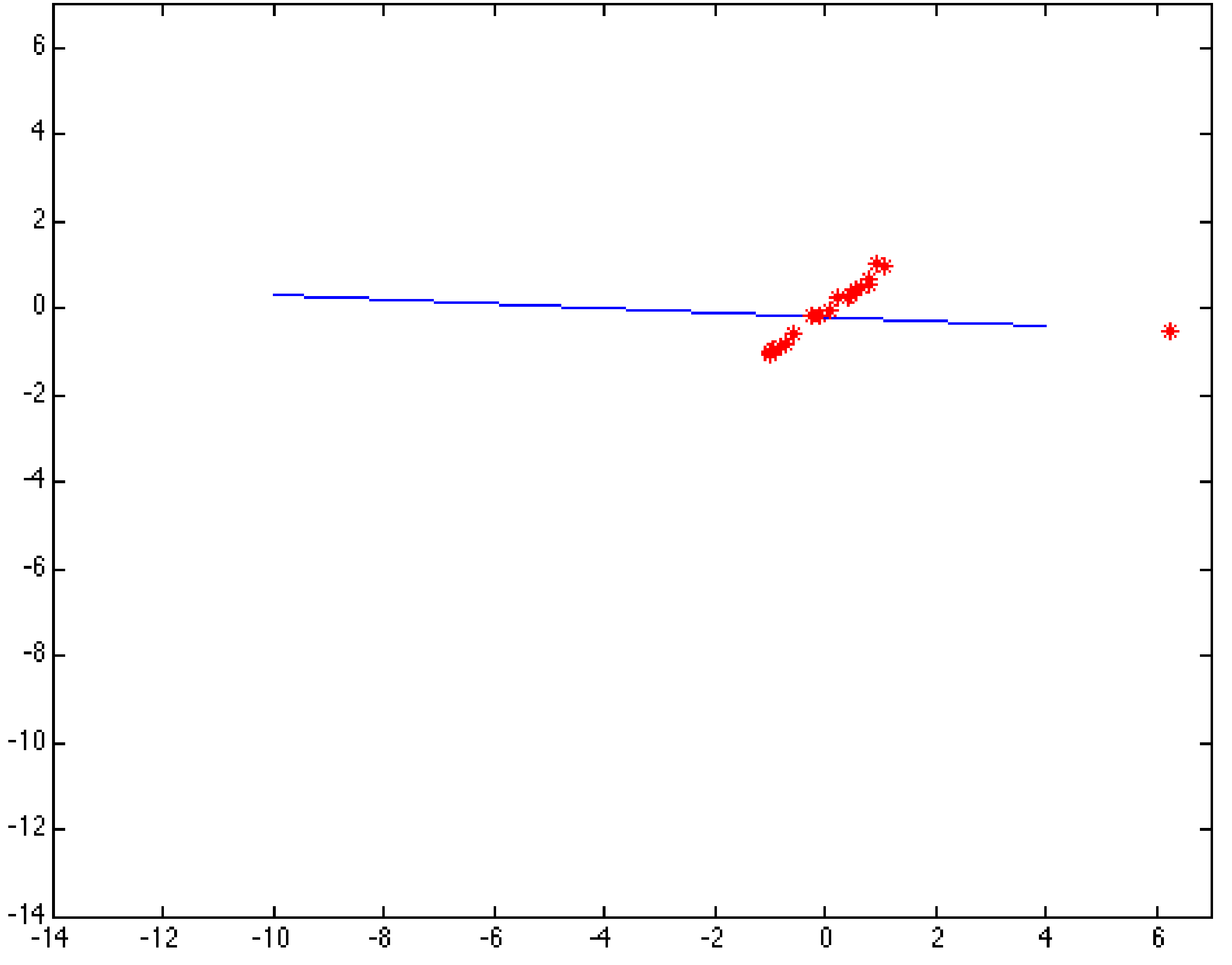
(2) Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S

(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

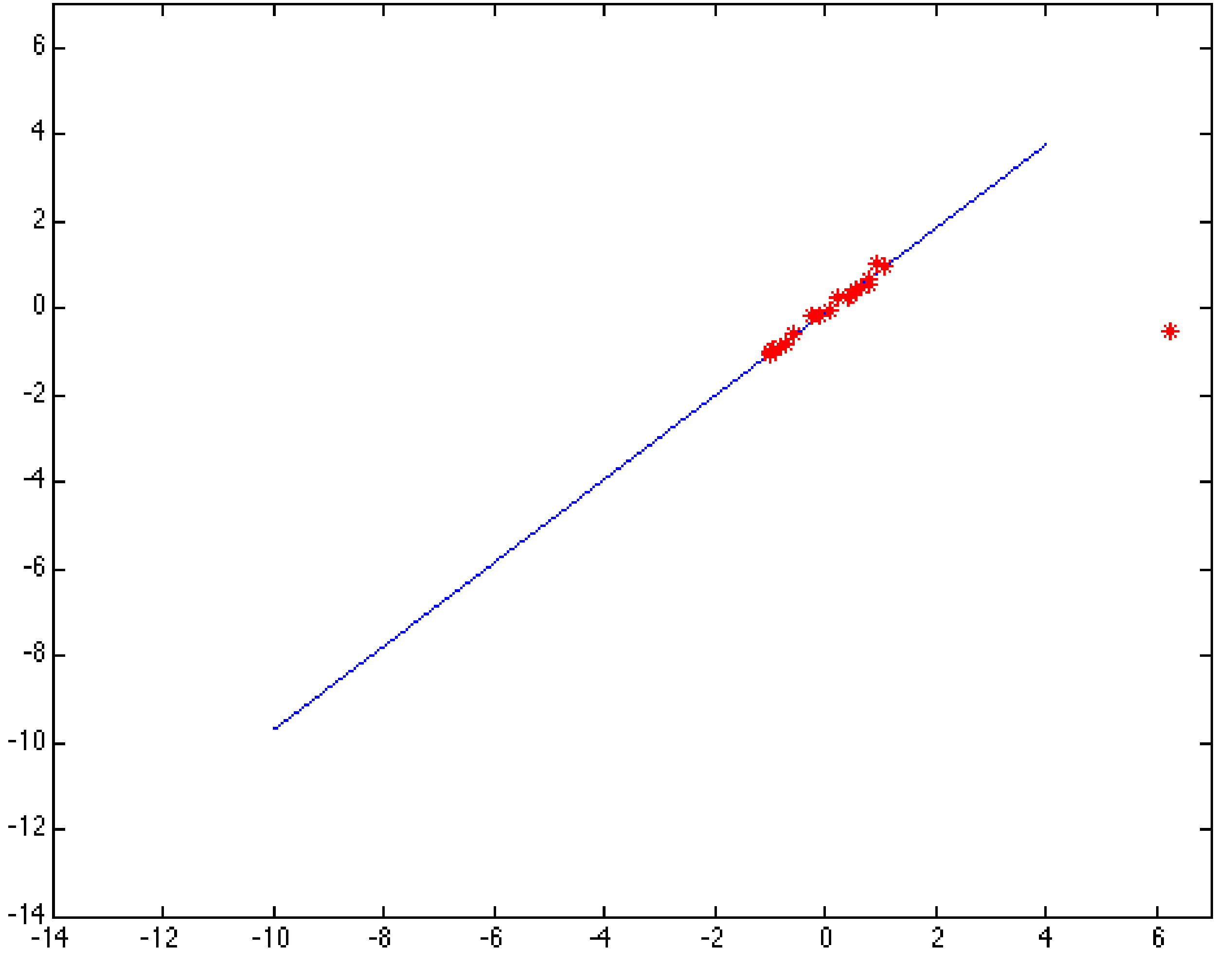
(4) If the size of S_i is less than T, select a new subset and repeat the above

(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

RANSAC-based Homography Estimation

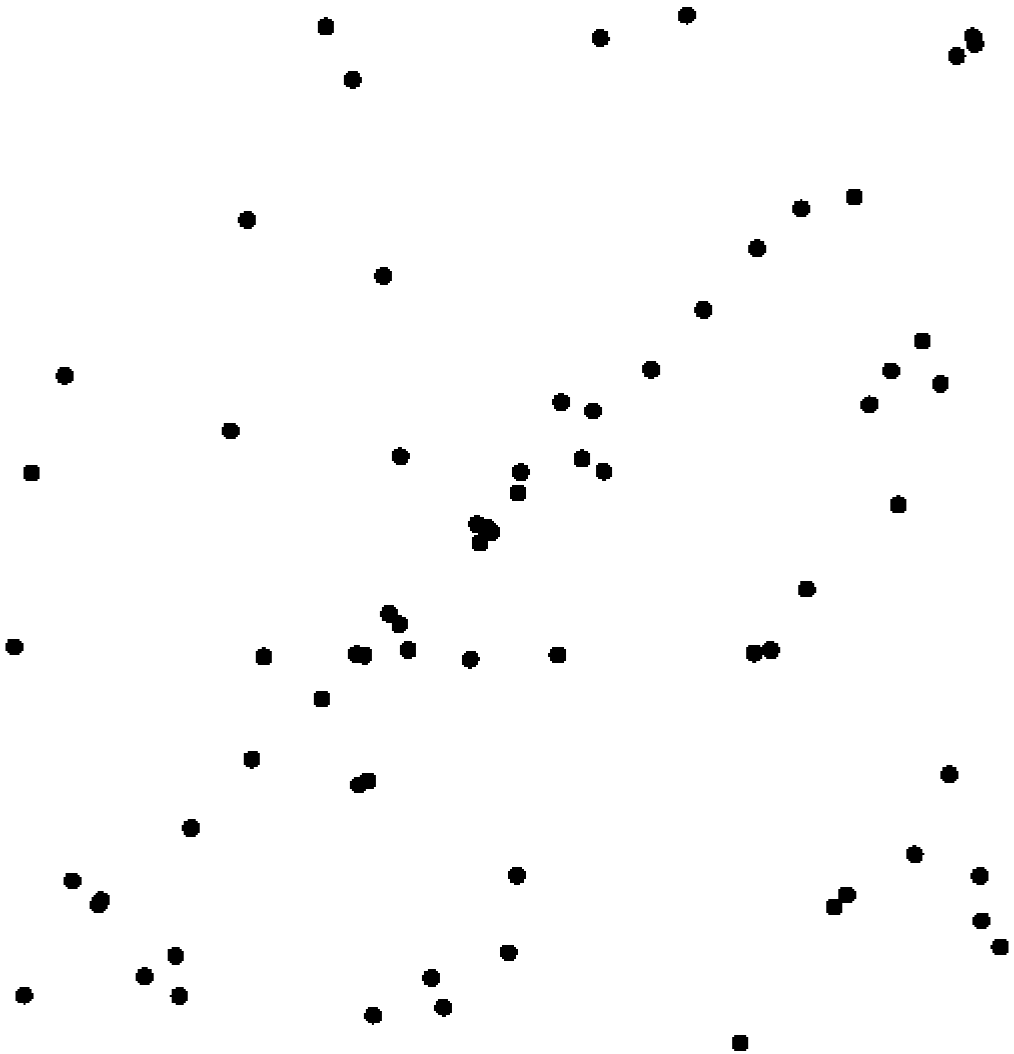


RANSAC-based Homography Estimation



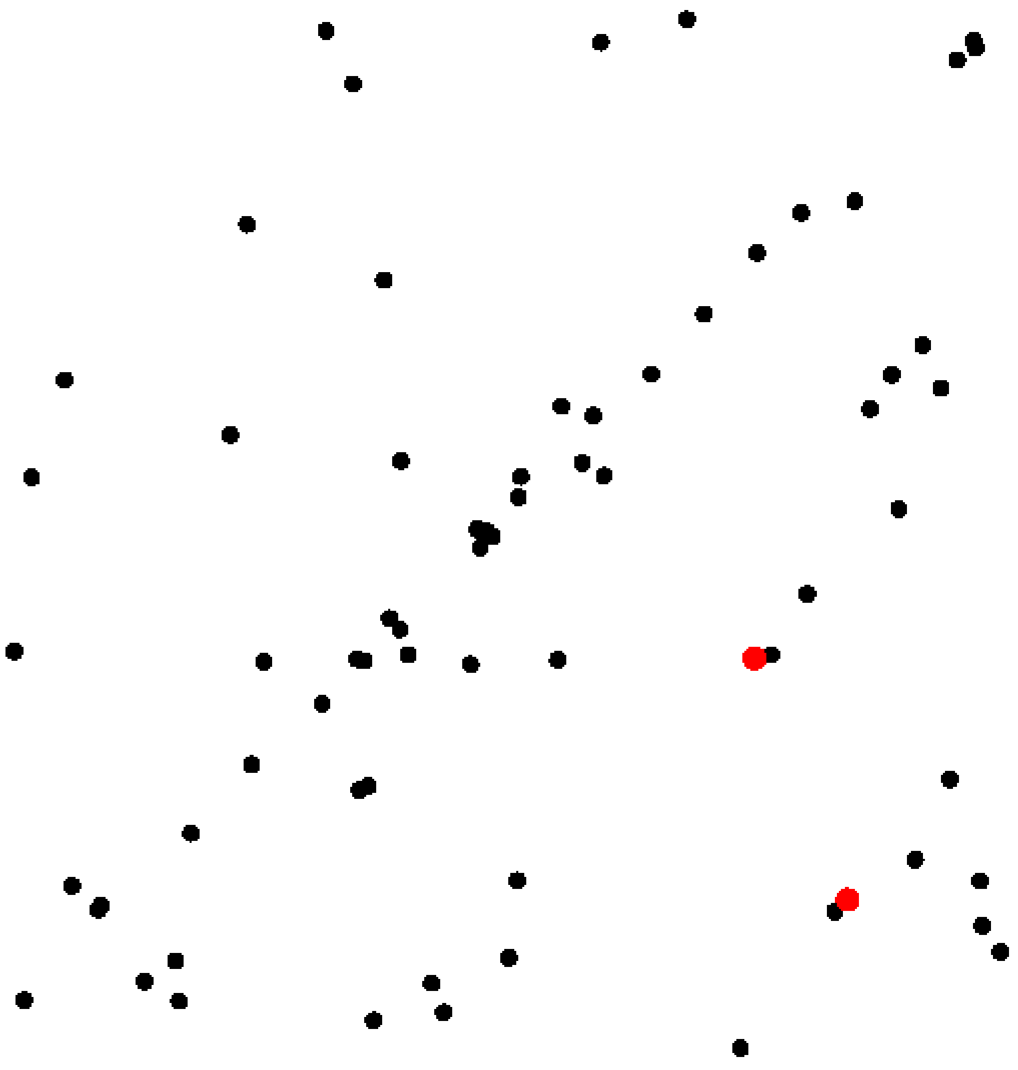
RANSAC-based Homography Estimation

Line fitting by RANSAC



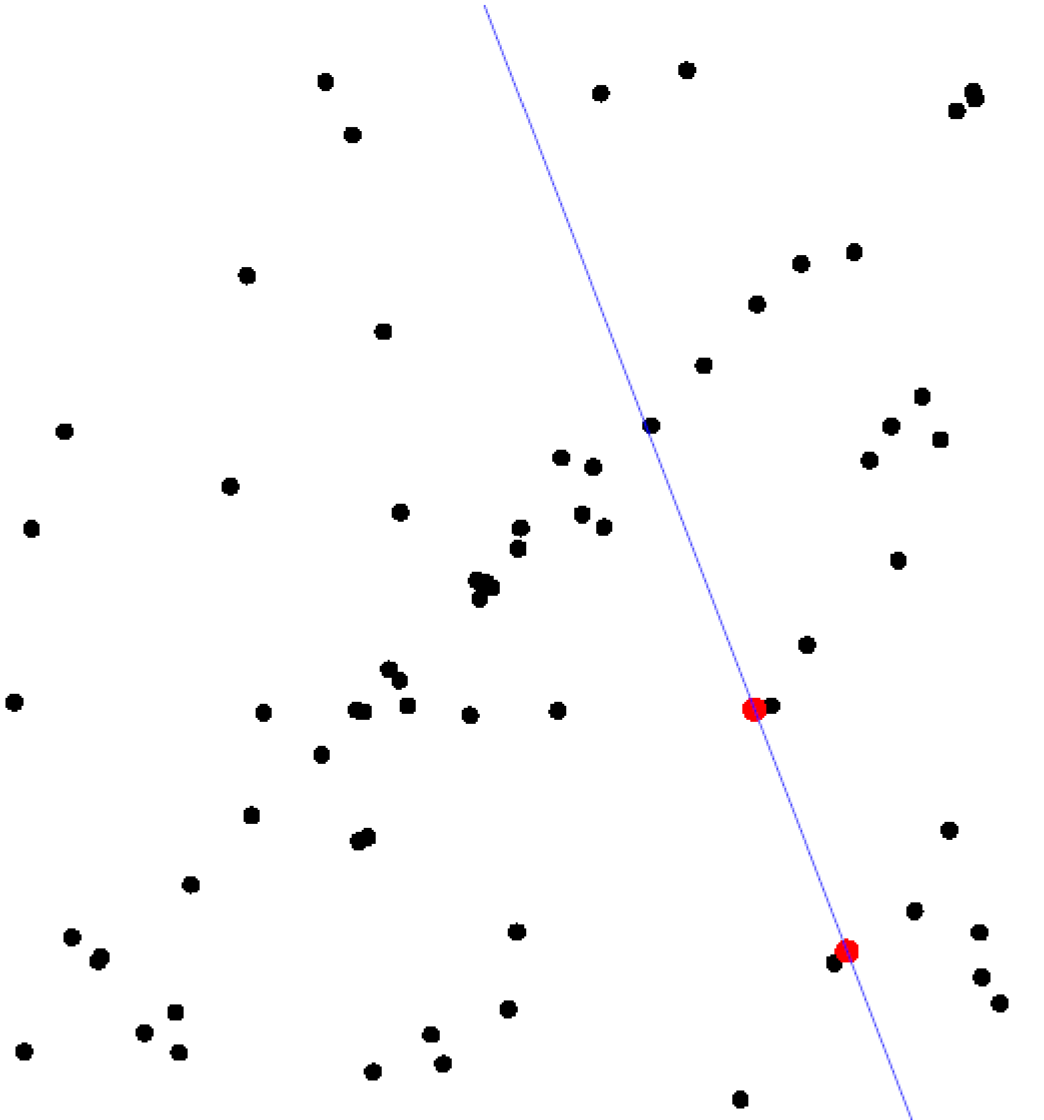
RANSAC-based Homography Estimation

Line fitting by RANSAC



- Randomly select two points

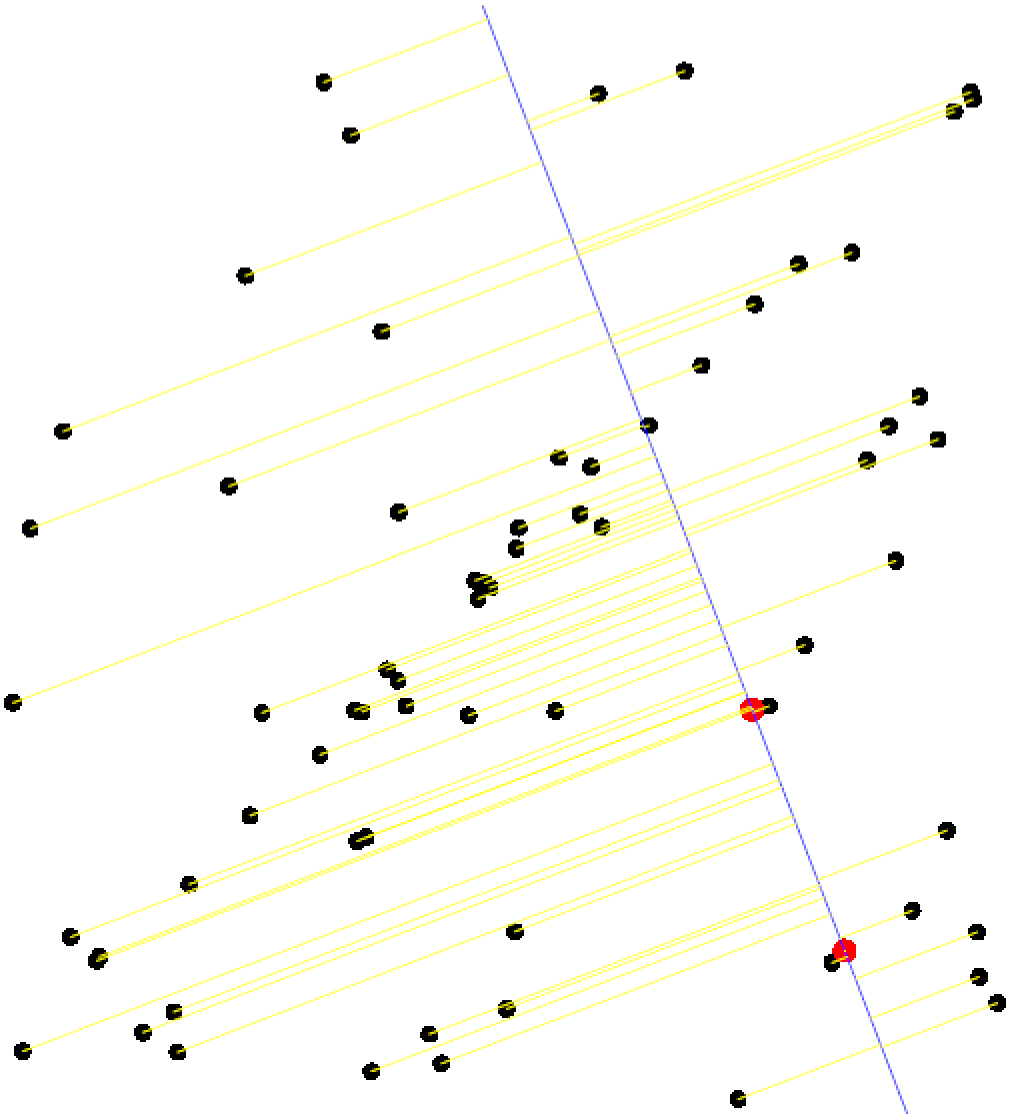
RANSAC-based Homography Estimation



Line fitting by RANSAC

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

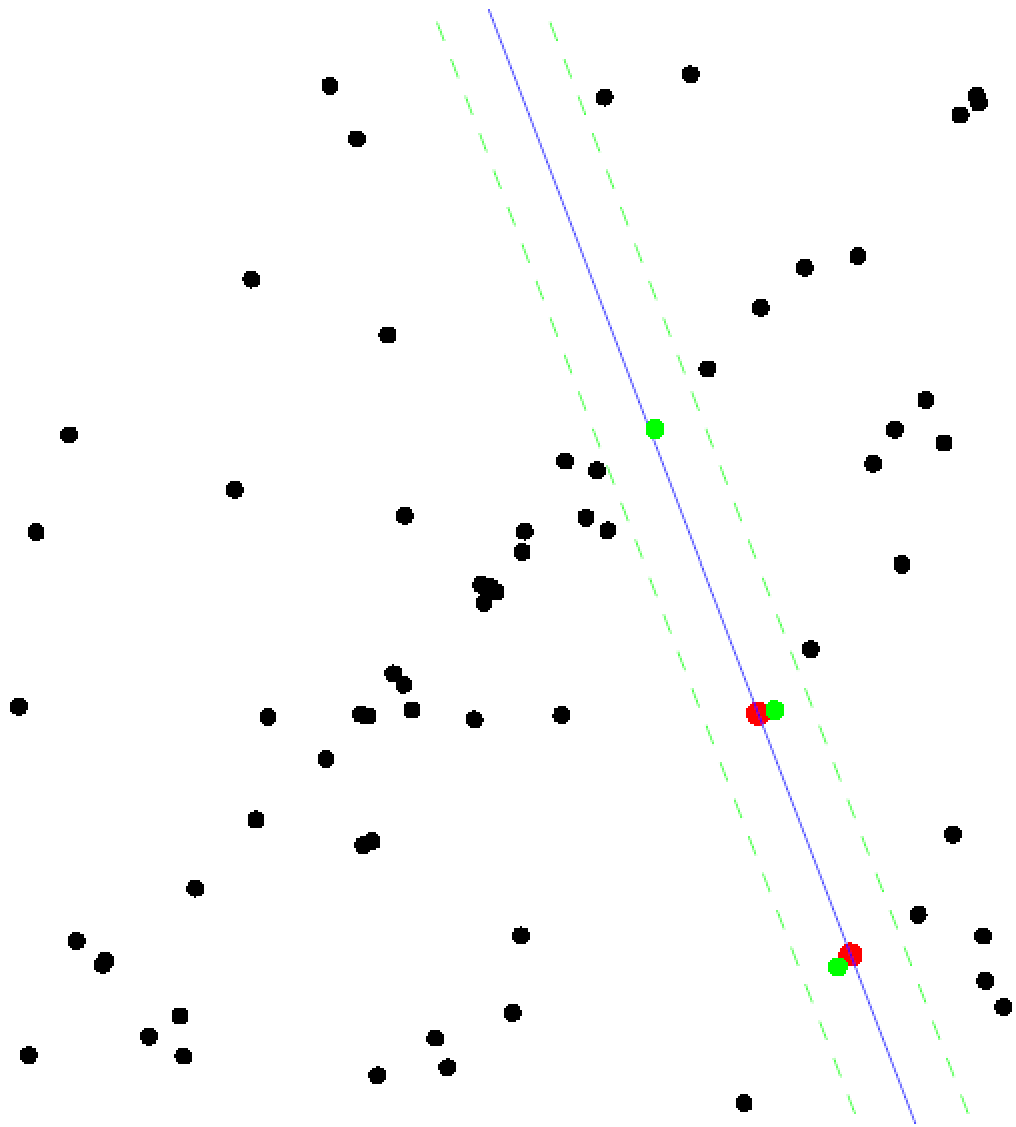
RANSAC-based Homography Estimation



Line fitting by RANSAC

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

RANSAC-based Homography Estimation

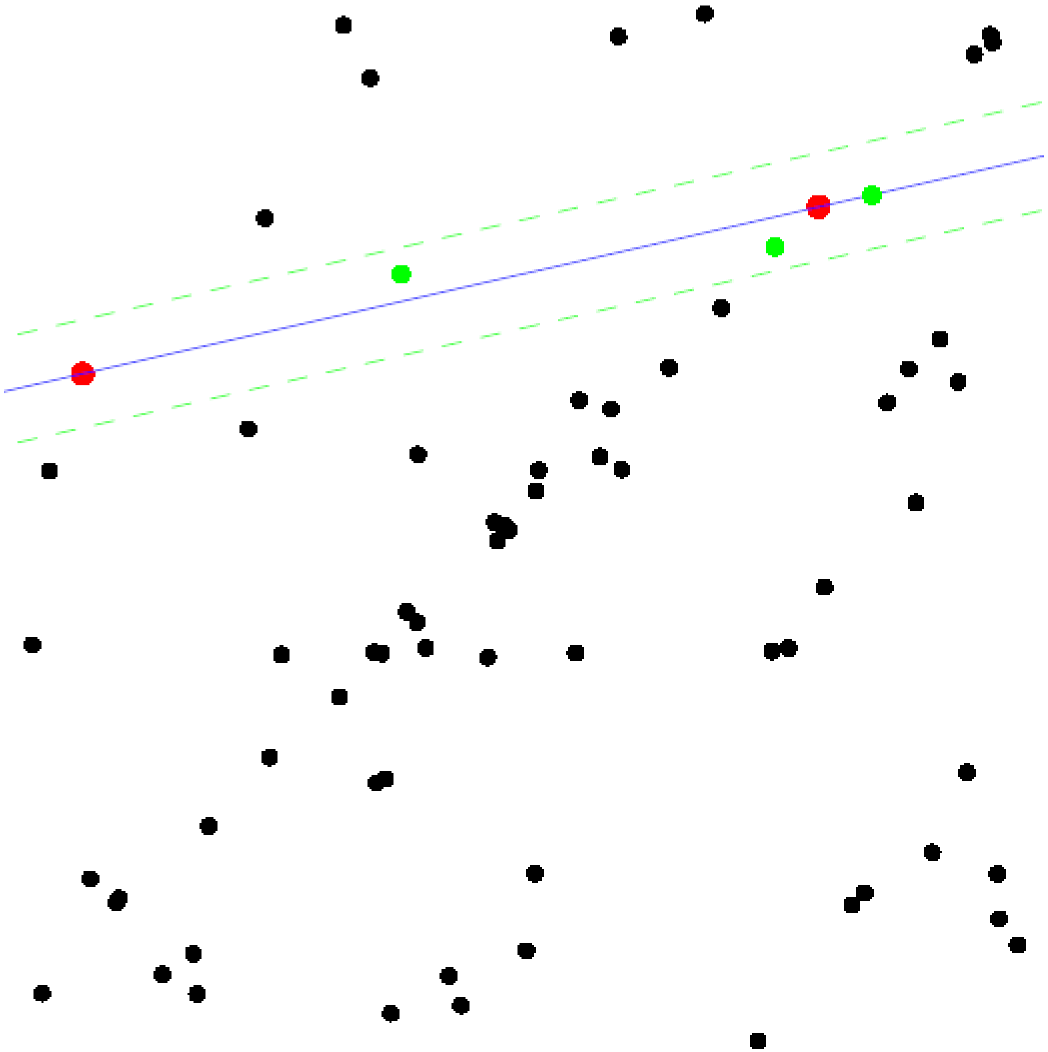


Line fitting by RANSAC

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

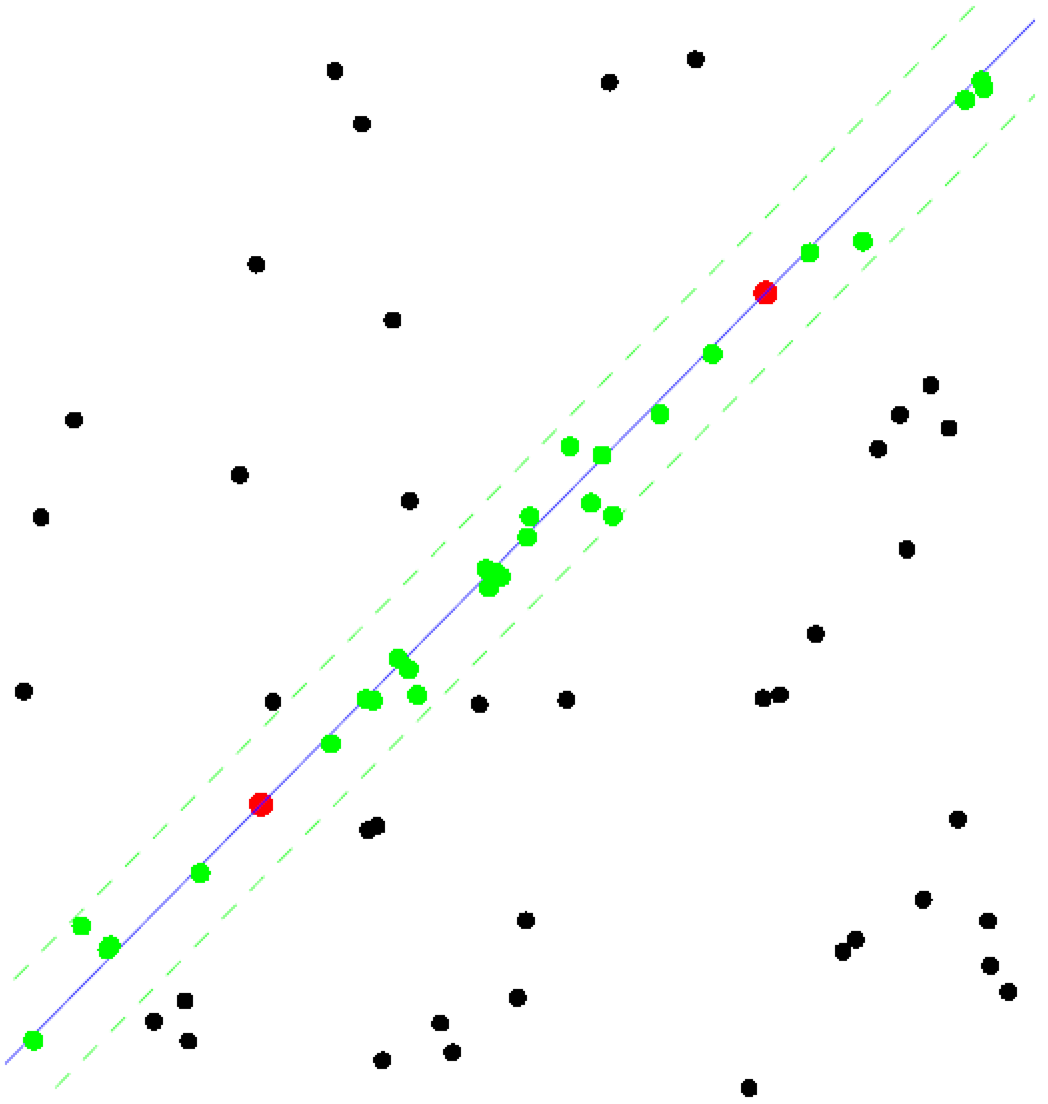
RANSAC-based Homography Estimation

Line fitting by RANSAC



- Test another two points

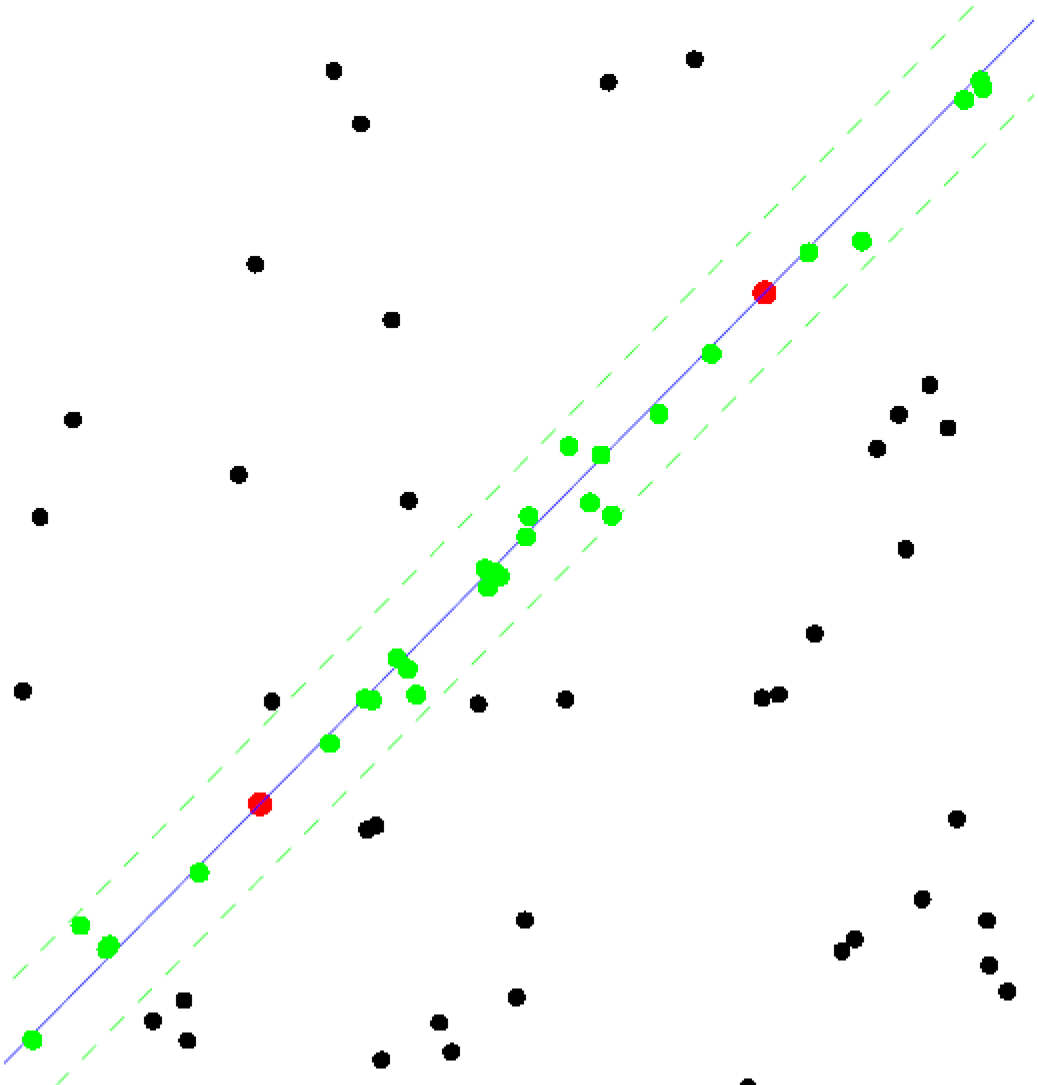
RANSAC-based Homography Estimation



Line fitting by RANSAC

- The final fitting result

RANSAC-based Homography Estimation



Line fitting by RANSAC

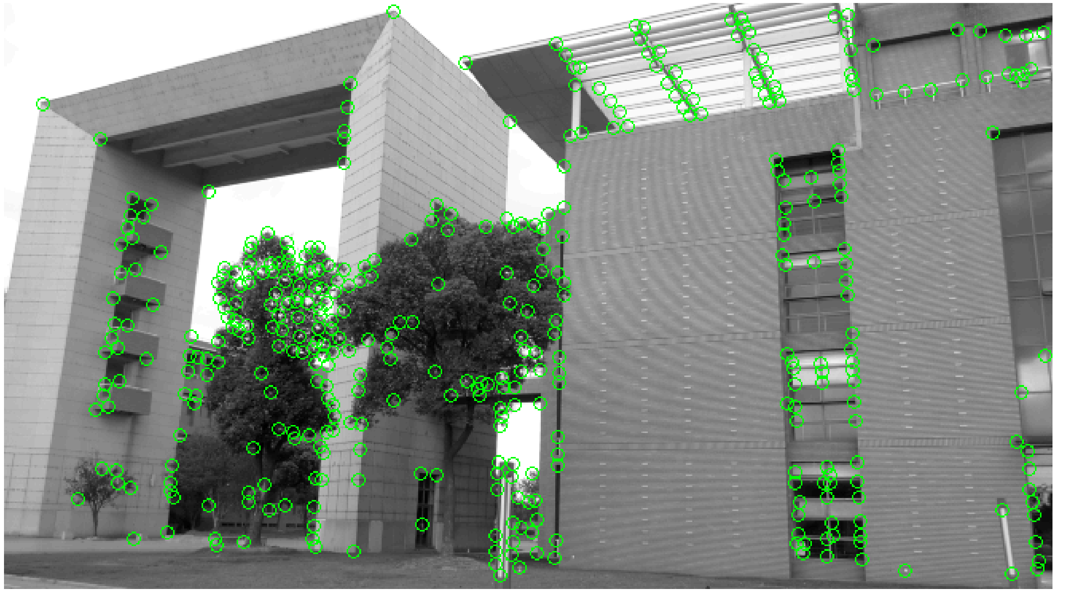
- The final fitting result

RANSAC-based Homography Estimation

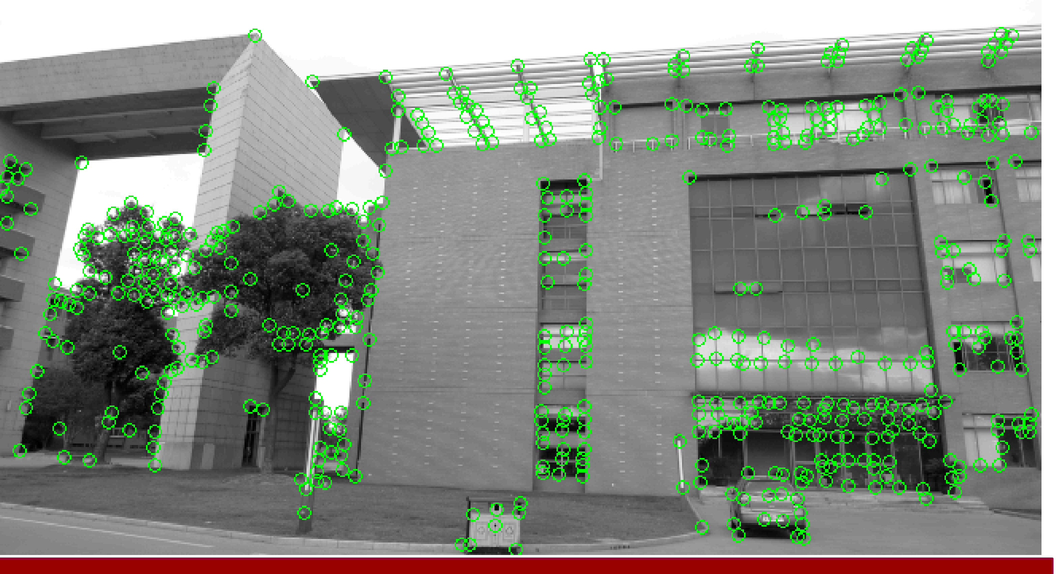
Can you describe the steps of homography estimation when using RANSAC?

Homography Estimation: Example 1

Homography Estimation: Example 1



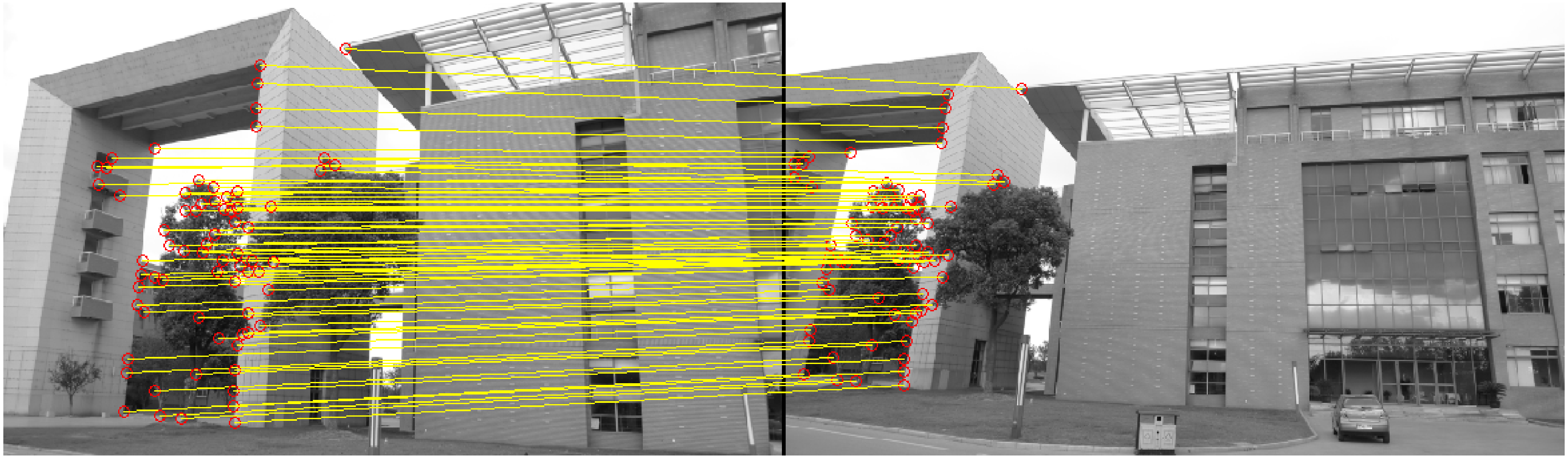
Interest points detection



Homography Estimation: Example 1

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

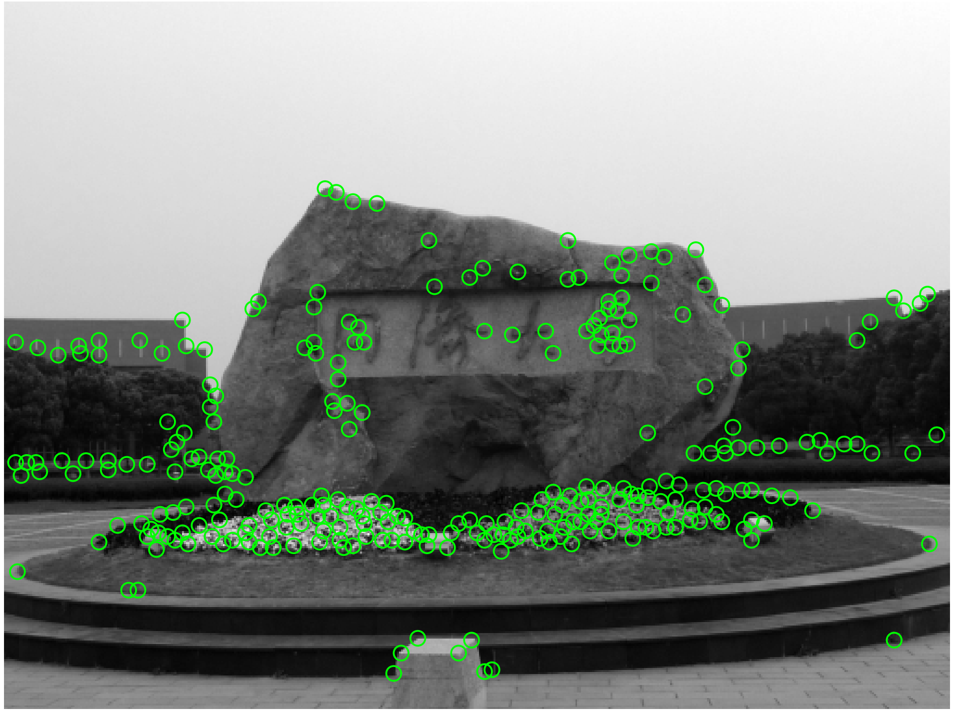
Homography Estimation: Example 1

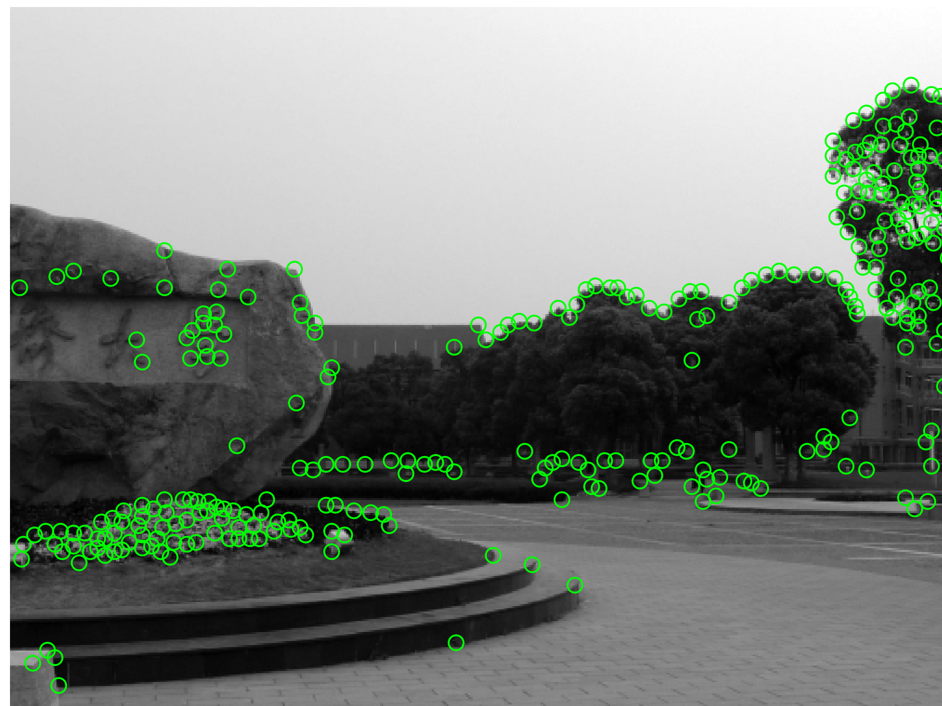
Finally, stitch the transformed image one with image two

Homography Estimation: Example 2

Homography Estimation: Example 2

Interest points detection

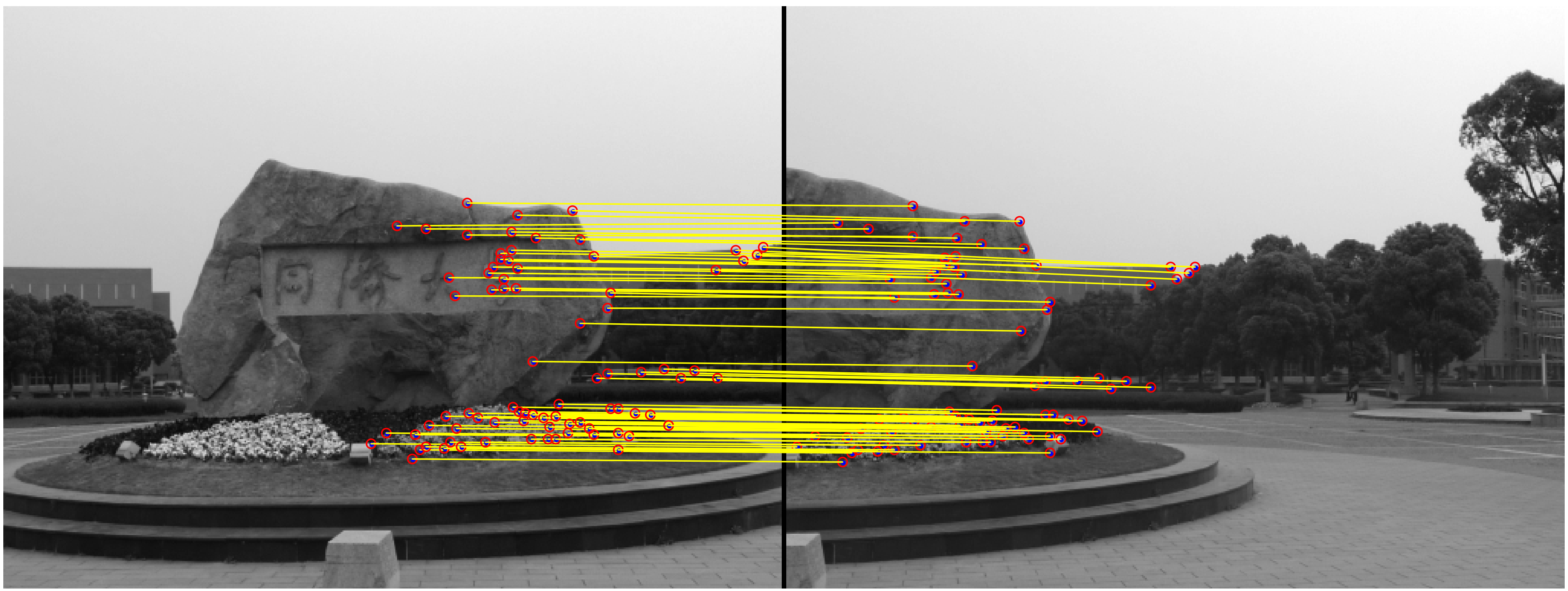




Homography Estimation: Example 2

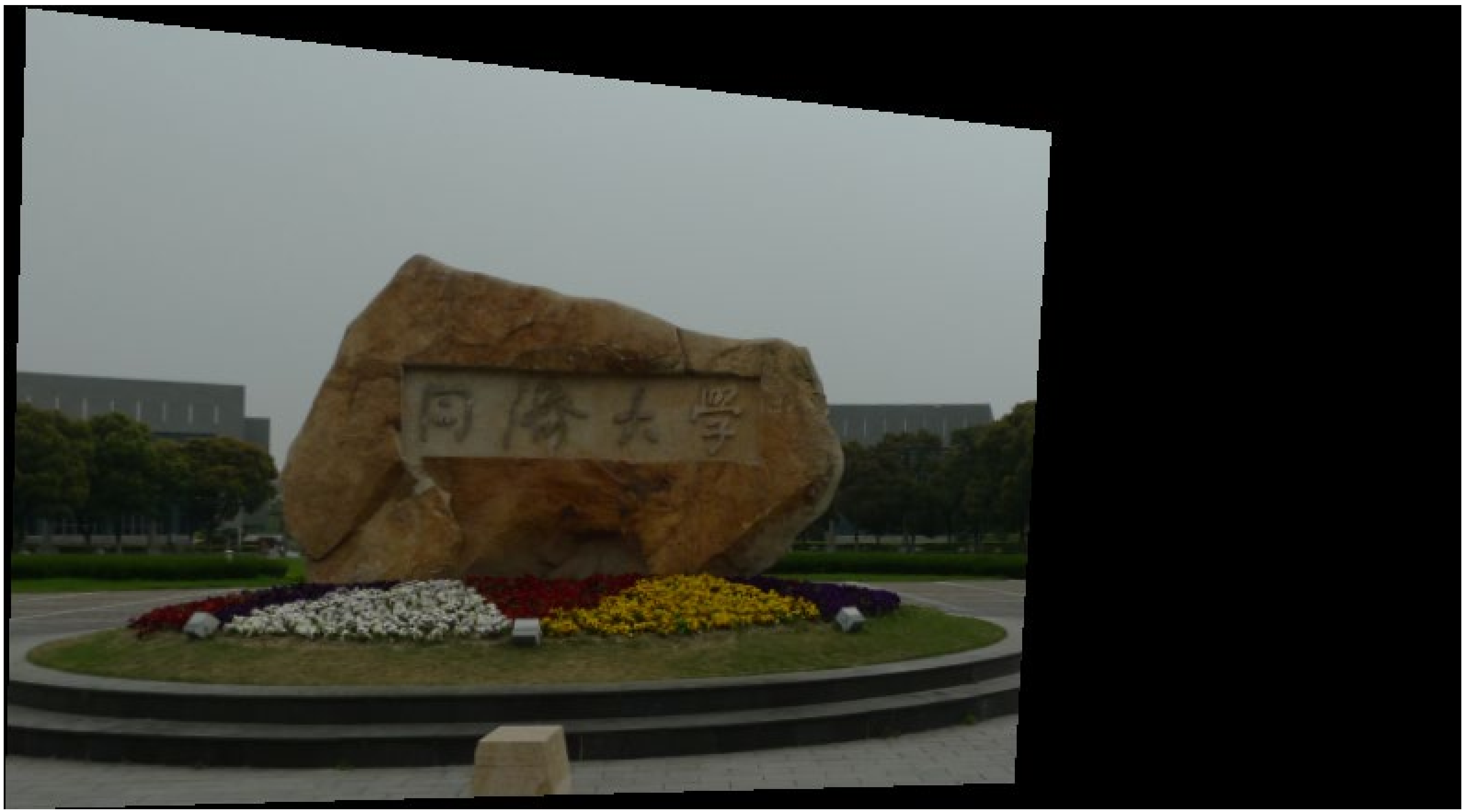
Correspondence estimation

Then, the homography matrix can be estimated by using the correspondence pairs with RANSAC



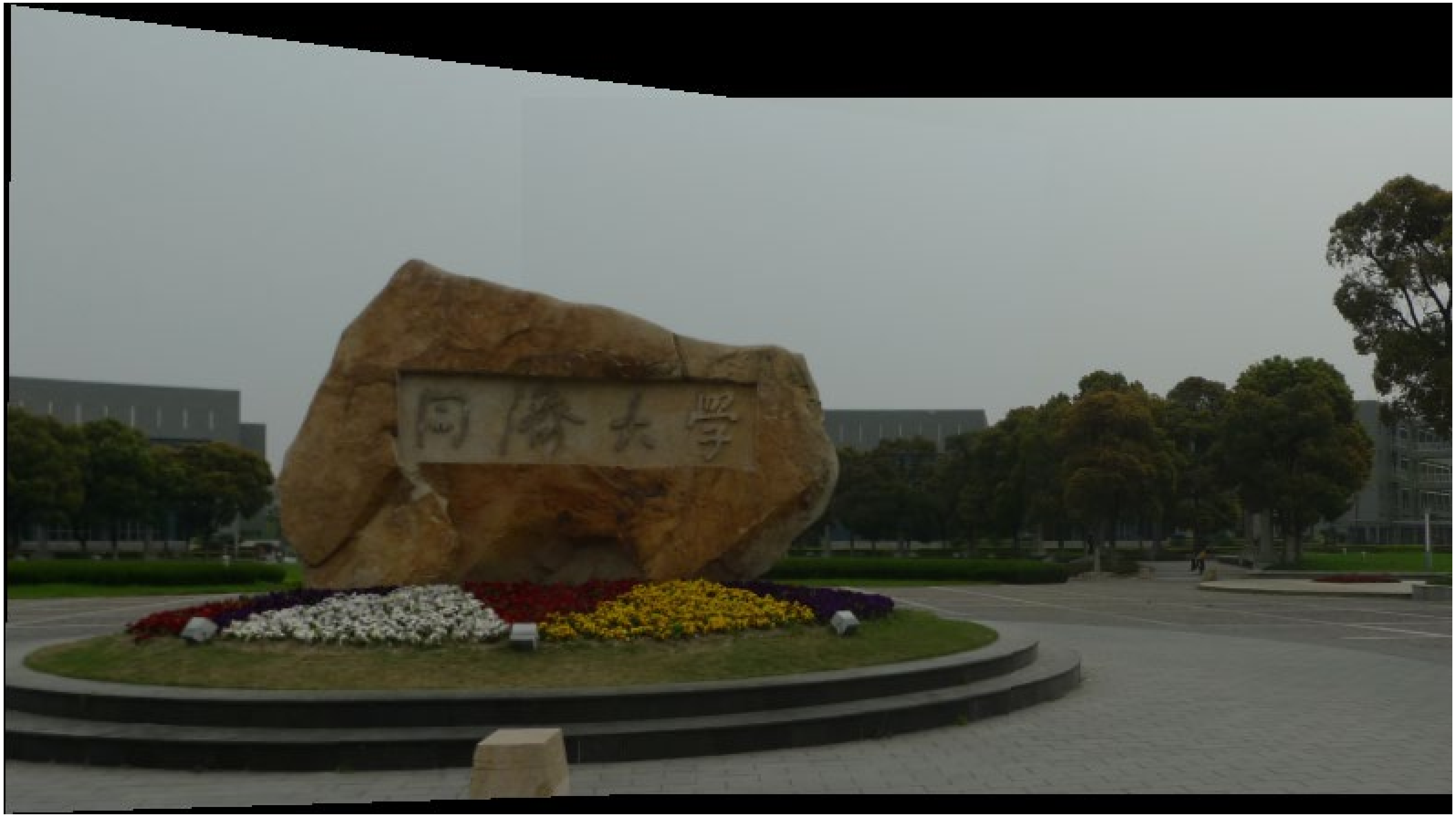
Homography Estimation: Example 2

Transform image one using the estimated homography matrix



Homography Estimation: Example 2

Finally, stitch the transformed image one with image two



Homography Estimation: Example 3

Project products of students from 2009 Media&Arts