Assignment1

> Computer Vision 2023 Spring

1953921 陈元哲

1

(Math) In our lectures, we mentioned that matrices that can represent Euclidean transformations can form a group. Specifically, in 3D space, the set comprising matrices $\{M_i\}$ is actually a group, where

$M_i = \begin{bmatrix} R_i & \mathbf{t}_i \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}$, $R_i \in \mathbb{R}^{3\times3}$ is an orthonormal matrix, det($\mathbf{R}_i$)=1, and $\mathbf{t}_i \in \mathbb{R}^{3\times1}$ is a vector.

Please prove that the set $\{M_i\}$ forms a group.

Hint: You need to prove that $\{M_i\}$ satisfies the four properties of a group, i.e., the closure, the associativity, the existence of an identity element, and the existence of an inverse element for each group element.

1. Closure

   First, we prove that for any two elements in the set, their product must be in the set

$$\forall \boldsymbol{M}_j, \boldsymbol{M}_k \in \{\boldsymbol{M}_i\}$$

$$\boldsymbol{M}_j\boldsymbol{M}_k = \begin{bmatrix} \boldsymbol{R}_j & \mathbf{t}_\mathrm{j} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_k & \mathbf{t}_\mathrm{k} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_j\boldsymbol{R}_k & \boldsymbol{R}_j\mathbf{t}_\mathrm{k} + \mathbf{t}_\mathrm{j} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix}$$

$$\because \boldsymbol{R}_j\boldsymbol{R}_k \in \mathbb{R}^{3\times3}, \boldsymbol{R}_j\mathbf{t}_\mathrm{k} + \mathbf{t}_\mathrm{j} \in \mathbb{R}^{3\times1}, \det(\boldsymbol{R}_j\boldsymbol{R}_k) = \det(\boldsymbol{R}_j)\det(\boldsymbol{R}_k) = 1$$

$$\therefore \boldsymbol{M}_j\boldsymbol{M}_k \in \{\boldsymbol{M}_i\}$$

2. Associativity

   Then, we prove that the order in which operations are performed does not matter.

$$\forall \boldsymbol{M}_j, \boldsymbol{M}_k, \boldsymbol{M}_l \in \{\boldsymbol{M}_i\}$$

$$(\boldsymbol{M}_j\boldsymbol{M}_k)\boldsymbol{M}_l = \begin{bmatrix} \boldsymbol{R}_j\boldsymbol{R}_k & \boldsymbol{R}_j\mathbf{t}_\mathrm{k} + \mathbf{t}_\mathrm{j} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_l & \mathbf{t}_\mathrm{l} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_j\boldsymbol{R}_k\boldsymbol{R}_l & \boldsymbol{R}_j\boldsymbol{R}_k\mathbf{t}_\mathrm{l} + \boldsymbol{R}_j\mathbf{t}_\mathrm{k} + \mathbf{t}_\mathrm{j} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{M}_j(\boldsymbol{M}_k\boldsymbol{M}_l) = \begin{bmatrix} \boldsymbol{R}_j & \mathbf{t}_\mathrm{j} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_k\boldsymbol{R}_l & \boldsymbol{R}_k\mathbf{t}_\mathrm{l} + \mathbf{t}_\mathrm{k} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_j\boldsymbol{R}_k\boldsymbol{R}_l & \boldsymbol{R}_j\boldsymbol{R}_k\mathbf{t}_\mathrm{l} + \boldsymbol{R}_j\mathbf{t}_\mathrm{k} + \mathbf{t}_\mathrm{j} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix}$$

$$(\boldsymbol{M}_j\boldsymbol{M}_k)\boldsymbol{M}_l = \boldsymbol{M}_j(\boldsymbol{M}_k\boldsymbol{M}_l)$$

3. Identity

   We need to find a matrix from the set which have the properties of identity. consider the matrix below

$$\boldsymbol{I} = \begin{bmatrix} \boldsymbol{E}_3 & \mathbf{0} \\ \mathbf{0}^\mathrm{T} & \mathbf{1} \end{bmatrix} \in \boldsymbol{M}_i$$

   we have

$$\forall \boldsymbol{M}_j \in \{\boldsymbol{M}_i\}$$
$$\boldsymbol{M}_j\boldsymbol{I} = \begin{bmatrix} \boldsymbol{R}_j & \mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_3 & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \boldsymbol{M}_j = \begin{bmatrix} \boldsymbol{E}_3 & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_j & \mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \boldsymbol{I}\boldsymbol{M}_j$$

So, $\boldsymbol{I}$ is the identity of this set.

4. Inverse

We prove for any element in the set, we can construct a matrix satisfied the properties of inverse.

$$\forall \boldsymbol{M}_j \in \{\boldsymbol{M}_i\}$$
$$\boldsymbol{M}_j = \begin{bmatrix} \boldsymbol{R}_j & \mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$$

we construct the matrix below

$$\because \det(\boldsymbol{R}_j) = 1 \neq 0$$
$$\therefore \exists \boldsymbol{R}_j^{-1}, \det(\boldsymbol{R}_j^{-1}) = 1 \quad s.t. \quad \boldsymbol{R}_j\boldsymbol{R}_j^{-1} = \boldsymbol{R}_j^{-1}\boldsymbol{R}_j = \boldsymbol{E}$$
$$\exists \boldsymbol{M}_j^{-1} = \begin{bmatrix} \boldsymbol{R}_j^{-1} & -\boldsymbol{R}_j^{-1}\mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \{\boldsymbol{M}_i\}$$

then, we prove $\boldsymbol{M}_j^{-1}$ is inverse of $\boldsymbol{M}_j$

$$\boldsymbol{M}_j\boldsymbol{M}_j^{-1} = \begin{bmatrix} \boldsymbol{R}_j & \mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_j^{-1} & -\boldsymbol{R}_j^{-1}\mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_j\boldsymbol{R}_j^{-1} & -\boldsymbol{R}_j\boldsymbol{R}_j^{-1}\mathbf{t_j} + \mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & -\boldsymbol{E}\mathbf{t_j} + \mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \boldsymbol{I}$$
$$\boldsymbol{M}_j^{-1}\boldsymbol{M}_j = \begin{bmatrix} \boldsymbol{R}_j^{-1} & -\boldsymbol{R}_j^{-1}\mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_j & \mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_j^{-1}\boldsymbol{R}_j & \boldsymbol{R}_j^{-1}\mathbf{t_j} - \boldsymbol{R}_j^{-1}\mathbf{t_j} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \boldsymbol{I}$$
$$\because \boldsymbol{M}_j\boldsymbol{M}_j^{-1} = \boldsymbol{M}_j^{-1}\boldsymbol{M}_j = \boldsymbol{I} \therefore \boldsymbol{M}_j^{-1} \; is \; inverse \; of \; \boldsymbol{M}_j$$

Based on the four properties mentioned above, the set $\{\boldsymbol{M}_i\}$ forms a group.

2

(Math) When deriving the Harris corner detector, we get the following matrix M composed of first-order partial derivatives in a local image patch w,

$$M = \begin{bmatrix} \sum_{(x_i,y_i)\in w} (I_x)^2 & \sum_{(x_i,y_i)\in w} (I_x I_y) \\ \sum_{(x_i,y_i)\in w} (I_x I_y) & \sum_{(x_i,y_i)\in w} (I_y)^2 \end{bmatrix}$$

a) Please prove that M is positive semi-definite.

b) In practice, M is usually positive definite. If M is positive definite, prove that in the Cartesian coordinate system, $[x, y]M\begin{bmatrix} x \\ y \end{bmatrix} = 1$ represents an ellipse.

c) Suppose that M is positive definite and its two eigen-values are λ_1 and λ_2 and $\lambda_1 > \lambda_2 > 0$. For the ellipse defined by $[x, y]M\begin{bmatrix} x \\ y \end{bmatrix} = 1$, prove that the length of its semi-major axis is $\frac{1}{\sqrt{\lambda_2}}$ while the length of its semi-minor axis is $\frac{1}{\sqrt{\lambda_1}}$.

- a

$$\forall \boldsymbol{v} \in \mathbb{R}^{2\times 1} \quad \boldsymbol{v} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\boldsymbol{v^T M v} = [x \quad y] \begin{bmatrix} \sum_{(x_i,y_i)\in w}(I_x)^2 & \sum_{(x_i,y_i)\in w}(I_x I_y) \\ \sum_{(x_i,y_i)\in w}(I_x I_y) & \sum_{(x_i,y_i)\in w}(I_y)^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$= \left[x \sum_{(x_i,y_i)\in w}(I_x)^2 + y \sum_{(x_i,y_i)\in w}(I_x I_y) \quad x \sum_{(x_i,y_i)\in w}(I_x I_y) + y \sum_{(x_i,y_i)\in w}(I_y)^2\right] \begin{bmatrix} x \\ y \end{bmatrix}$$

$$= x^2 \sum_{(x_i,y_i)\in w}(I_x)^2 + xy \sum_{(x_i,y_i)\in w}(I_x I_y) + xy \sum_{(x_i,y_i)\in w}(I_x I_y) + y^2 \sum_{(x_i,y_i)\in w}(I_y)^2$$

$$= \sum_{(x_i,y_i)\in w} x^2(I_x)^2 + 2xy(I_x I_y) + y^2(I_y)^2$$

$$= \sum_{(x_i,y_i)\in w} (xI_x + yI_y)^2 \geq 0$$

According to the decision theorem, $\boldsymbol{M}$ is a positive semi-definite matrix.

- b

$\boldsymbol{M}$ is positive definite, we assume its eigen-values is $\lambda_1, \lambda_2 > 0$, their corresponding eigen-vectors are orthogonal, we record the normalized eigen-vectors as μ_1, μ_2, $A = [\mu_1 \ \mu_2]$ is orthogonal matrix.

Accroding to positive definite matrix's properties

$$\boldsymbol{M} = \boldsymbol{A} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \boldsymbol{A^T}$$

let $\boldsymbol{v} = \begin{bmatrix} x \\ y \end{bmatrix}$, we have

$$\boldsymbol{v^T M v} = \boldsymbol{v^T A} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \boldsymbol{A^T v} = (\boldsymbol{A^T v})^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} (\boldsymbol{A^T v}) = 1$$

let $\boldsymbol{v'} = \begin{bmatrix} x' \\ y' \end{bmatrix}, \boldsymbol{v'} = (\boldsymbol{A^T v})$

$$\boldsymbol{v'}^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \boldsymbol{v'} = [x' \quad y'] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \lambda_1 x'^2 + \lambda_2 y'^2 = 1$$

$\lambda_1, \lambda_2 > 0$, $\boldsymbol{v^T M v}$ represents an ellipse whose major and minor axes are eigen-vectors μ_1, μ_2.

- c

We assume the length of the ellipse's semi-major axis is a , the length of semi-minor axis is b. Accroding to question b, we have the ellipse's expression

$$\lambda_1 x'^2 + \lambda_2 y'^2 = \frac{x'^2}{\frac{1}{\lambda_1}} + \frac{y'^2}{\frac{1}{\lambda_2}} = 1$$

$$a^2 = \frac{1}{\lambda_2}, b^2 = \frac{1}{\lambda_2}$$

$$a = \frac{1}{\sqrt{\lambda_2}}, b = \frac{1}{\sqrt{\lambda_1}}$$

3

(Math) In the lecture, we talked about the least square method to solve an over-determined linear system $A\mathbf{x} = b, A \in \mathbb{R}^{m\times n}, \mathbf{x} \in \mathbb{R}^{n\times 1}, m > n, rank(A) = n$. The closed form solution is $\mathbf{x} = \left(A^T A\right)^{-1} A^T b$. Try to prove that $A^T A$ is non-singular (or in other words, it is invertible).

First, we prove that $\boldsymbol{A}^T\boldsymbol{A}$ is a symmetric matrix

$$(\boldsymbol{A}^T\boldsymbol{A})^T = \boldsymbol{A}^T(\boldsymbol{A}^T)^T = \boldsymbol{A}^T\boldsymbol{A}$$

for any non-zero vector $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$

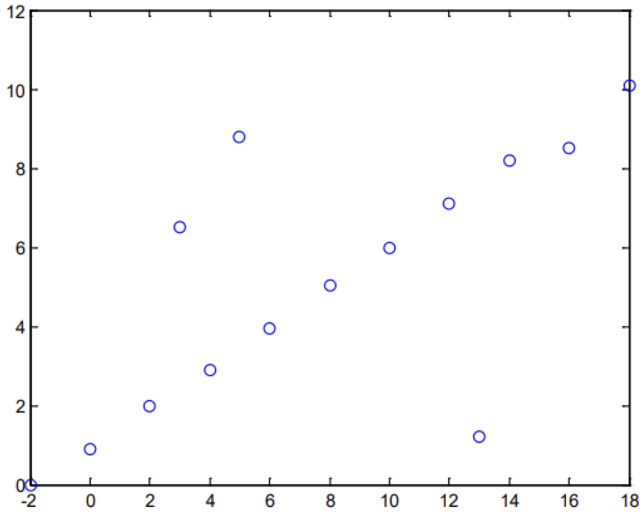
$$\because rank(\boldsymbol{A}) = n$$
$$\therefore \boldsymbol{A}\boldsymbol{x} \neq 0$$
$$\because (\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x} > 0$$
$$\therefore \boldsymbol{A}^T\boldsymbol{A}\, is\ positive\ definite$$

According to positive definite matrix's properties, $\boldsymbol{A}^T\boldsymbol{A}$ is non-singular.

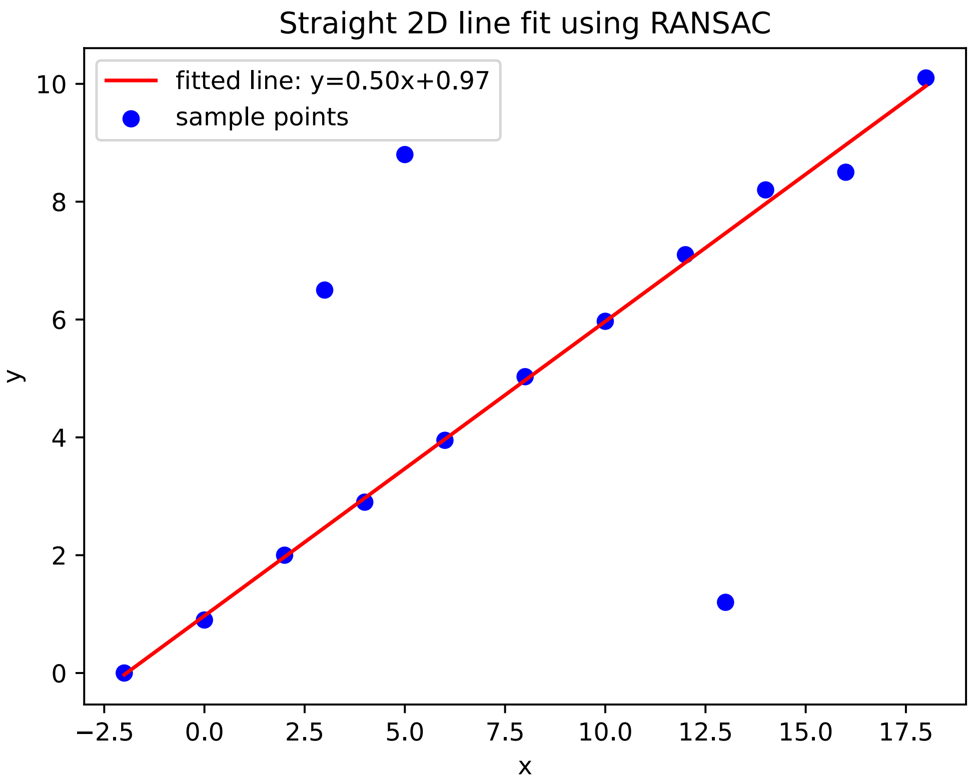
4

(Programming) RANSAC is widely used in fitting models from sample points with outliers. Please implement a program to fit a straight 2D line using RANSAC from the following sample points:

(-2, 0), (0, 0.9), (2, 2.0), (3, 6.5), (4, 2.9), (5, 8.8), (6, 3.95), (8, 5.03), (10, 5.97), (12, 7.1), (13, 1.2), (14, 8.2), (16, 8.5) (18, 10.1). Please show your result graphically.



We use python language to implement the program, having the result below, the source code is file `RANSAC.py` .

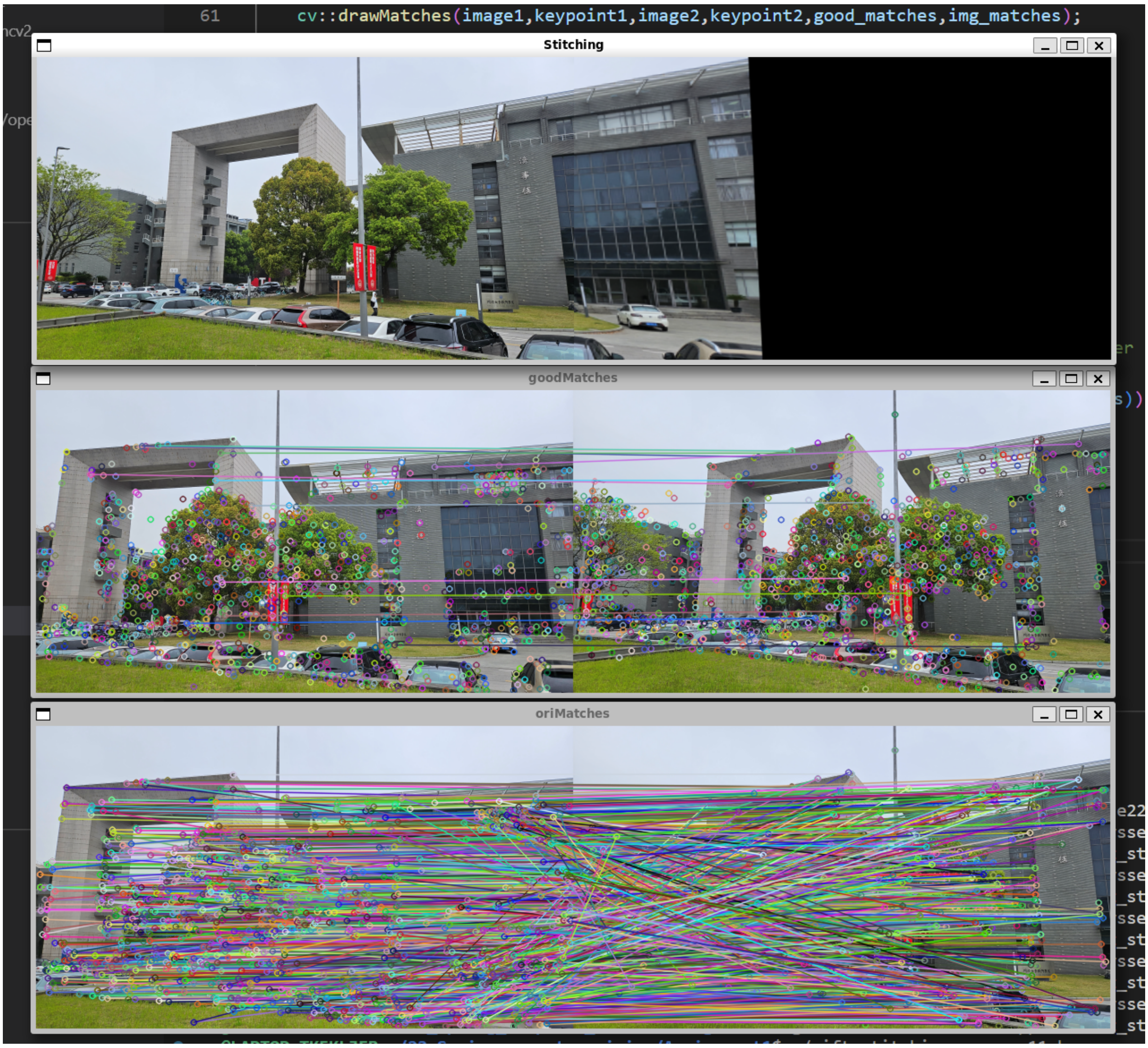


5

(Programming) Get two images I_1 and I_2 of our campus and make sure that the major parts of I_1 and I_2 are from the same physical plane. Stitch I_1 and I_2 together to get a panorama view using scale-normalized LoG (or DoG) based interest point detector and SIFT descriptors. You can use OpenCV or Matlab.

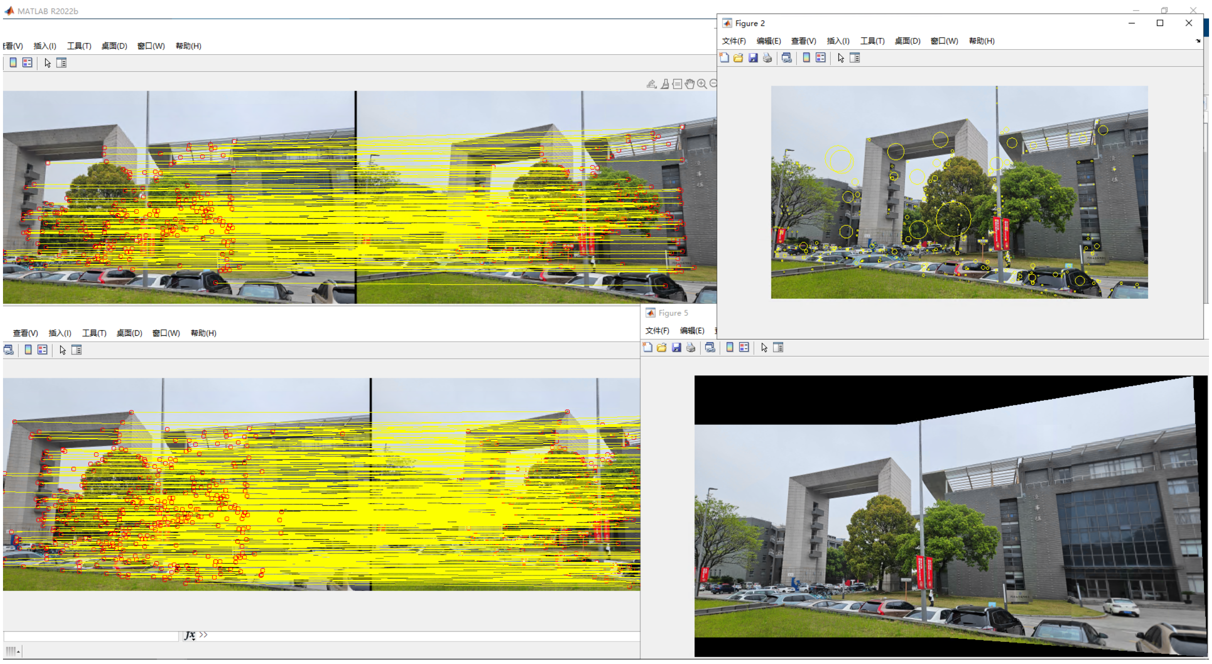
We use the OpenCV 3.4.19 in Linux to implement the panorama stitching using DoG based interest point detector and SIFT descriptors. The program I designed receives two args from command line, the paths of the given images, output the feature point matching before and after filterating the bad matches and the stitch result.

I tested the program with the photos of SSE I took by myself, here is the result:



The source code is file `Stitch.cpp`

I also tested my photos using the MATLAB program provided by Pro.Zhang, here is the result:



Comparing the two results above, we can find that the number of good matches in my program is much less than Pro.Zhang's program. It may due to I set the filtering threshold to small. The stitching result are similar, which shows the correctness of my program.

To compile and run the code, we should execute the commands below:

```
g++ -std=c++11 Stitch.cpp -o sift_stitching.o `pkg-config --cflags --libs opencv4`
./sift_stitching.o mysse11.bmp mysse22.bmp
```

6

(Programming) ORB feature point detection and matching algorithms have been fully implemented in the OpenCV library. Please write a C++program that invokes the OpenCV library's algorithm library for ORB feature point detection and matching for two given images, and output feature point matching results similar to the following given example.

We use the OpenCV 3.4.19 in Linux to implement the ORB feature point detection and matching for two given images. The program I designed receives two args from command line, the paths of the given images, output the feature point matching results.

I tested the program with the photos of SSE I took by myself, here is the result:

The source code is file `ORB.cpp`

To compile and run the code, we should execute the commands below:

```
g++ -std=c++11 ORB.cpp -o orb_matching.o `pkg-config --cflags --libs opencv4`
./orb_matching.o mysse11.bmp mysse22.bmp
```