Computer Vision assignment 3

이재원

June 28, 2024

1 Least square line fitting

The optimal line fitting using least squares starts with finding the eigenvectors of the second momentum. The smallest eigenvector is the normal vector of the fitted line. Also, the fitted line should pass the mean of all data.

2 Another interpretation of least square line fitting

2.1 The vector v that maximizes variance

Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of Σ arranged in descending order (i.e., $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$), and let $x_1, x_2, \ldots, x_n$ be the corresponding orthonormal eigenvectors. Since Σ is symmetric, it has a complete set of orthonormal eigenvectors.

Any vector v can be expressed as a linear combination of the eigenvectors of Σ:

$$v = \sum_{i=1}^{n} c_i x_i$$

,where c_i are scalars.

$$\begin{aligned} v^T \Sigma v &= (\sum_{i=1}^{n} c_i x_i)^T \Sigma (\sum_{i=1}^{n} c_i x_i) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \lambda_j x_i^T x_j \\ &\quad \text{because the eigenvectors are orthonormal,} \\ &= \sum_{i=1}^{n} c_i^2 \lambda_i \end{aligned}$$

similarly, $v^T v = \sum_{i=1}^{n} c_i^2$.

The Rayleigh Quotient $R(v) = \frac{v^T \Sigma v}{v^T v}$ is maximized when the largest weight is on λ_1. This happens when v is aligned with x_1, the eigenvector corresponding to λ_1. if v = x_1,

$$R(v) = \frac{\lambda_1 x_1^T x}{x_1^T x} = \lambda_1$$

2.2 Relation between Σ and $U^T U$

$n\Sigma$ is the second momentum.

2.3 Relation between v and n

As the normal vector n is the smallest eigenvector, and v is the largest eigenvector, they are different eigenvectors. Also, as the matrix is similar, the eigenvectors are orthogonal, so n and v are orthogonal.

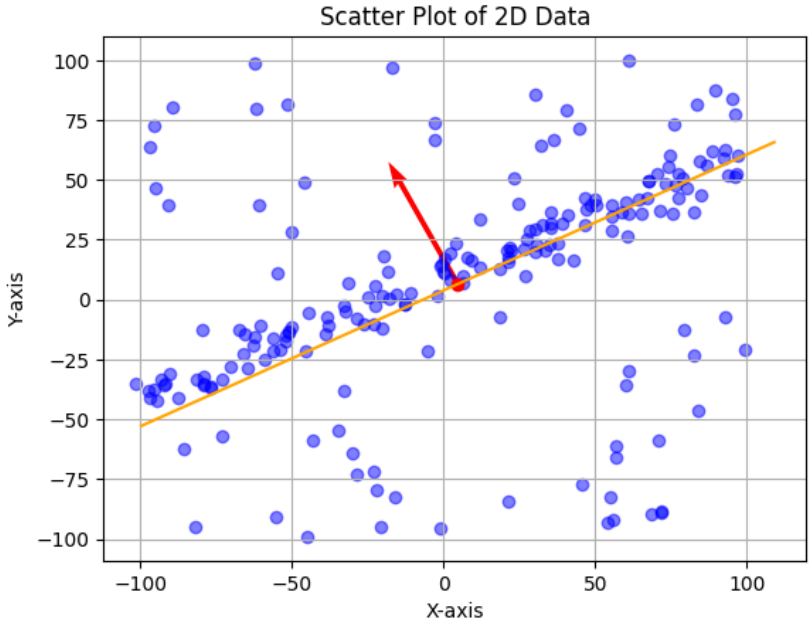

Figure 1: The optimal line that fits the data in total least square sense.

3 Homograpy and image stitching

3.1 Implementation

This is implemented by using *stich_right_image()* function twice. The *stich_right_image()* implements the image stiching of middle image and right image. So, you cannot stich the left image and middle image directly with *stich_right_image()*, so a trick is used. The trick is to use left/right inversion to the image, then the left and middle image becomes right and middle image. Then the result is left/right inversioned.

stich_right_image() is constructed of other functions. The functions are

- plot_figures()
- plot_all_matches()
- plot_match()
- find_matches()
- compute_homography_RANSAC()
- compute_homography_DLT()
- calculate_symmetric_transfer_error()
- find_inliner_matches()
- combine_images()
- warp_right_image()

3.2 Results

3.2.1 Middle image and Right image

Homography H23:

[-9.62483759e-03 -3.65325389e-04 -9.96902672e-01]

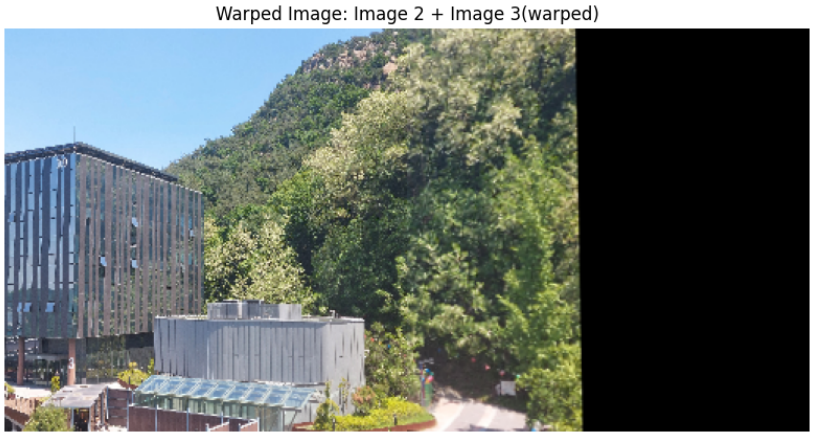

Figure 2: Image stitching of middle image and right image.

Figure 3: Image stitching of middle image and left image. (Left/right inversed)

[1.49321655e-03 -1.10890301e-02 -7.63502881e-02]
[8.55765623e-06 -4.57792893e-07 -1.17358012e-02]
total symmetric transfer error: 1127

3.2.2 Middle image and Left image

Homography H23:

[7.49251053e-03 2.97186950e-04 9.98825712e-01]
[-1.05011512e-03 8.56255979e-03 4.61801975e-02]
[-6.16559888e-06 -2.81208387e-07 9.16191567e-03]
total symmetric transfer error: 3253

4 Discussion

The final image had nice looking. The left image and the middle image is not stiched perfectly, but the middle image and the right image looks very natural. The reason why the left image and middle image are not stitched perfectly is because the homography was not correct. And this is because there are some noise to the images.

This image stiching can be inefficient because the homograpy was calculated 4 times (including left/right inversioned). This is because the warping function was implemented by using the prebuilt

Figure 4: The final image that combined all 3 images.

library, and the library only warped the right image to the reference left image. If using our own made warping function, homography can be only calculated 2 times.