

Homework #5

ECE661

Name: Chengzhang Zhong

Email: zhongc@purdue.edu

Step 1. SIFT corresponding points

Using the same procedures from homework 4 to implement SIFT.

In SIFT algorithm, we first need to find local extrema using DoG pyramid, which can be computed as:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma) * I(x, y)) = ff(x, y, k\sigma) - ff(x, y, \sigma)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

where we know $G(x, y, \sigma)$ is a variable scale Gaussian and $ff(x, y, \sigma)$ is Laplacian of Gaussian. And for each point in the DoG pyramid, it will need to be compared with 8 points at the same scale in 3x3 neighbor, 9 points in the next scale in 3x3 neighbor, 9 points in the previous scale in 3x3 neighbor.

To locate the extrema in the sub-pixel accuracy, we need to calculate the second derivative of $D(x, y, \sigma)$. To do this, we need to find the Taylor series expansion first:

$$D(x) = D(x_0) + J^T(x_0)x + \frac{1}{2}x^T H(x_0)x$$

where x is the incremental deviation of x_0 . J is the gradient vector and H is the Hessian matrix

$$J(x_0) = \left(\frac{\partial D}{\partial x}, \frac{\partial D}{\partial y}, \frac{\partial D}{\partial \sigma} \right)_{x_0}^T$$
$$H(x_0) = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial x \partial \sigma} \\ \frac{\partial^2 D}{\partial y \partial x} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y \partial \sigma} \\ \frac{\partial^2 D}{\partial \sigma \partial x} & \frac{\partial^2 D}{\partial \sigma \partial y} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix}$$

To threshold out weak extrema, we can set the threshold value equals to 0.03.

The remained points are named as candidate points.

To find the dominant local orientation, we need to calculate the gradient vector of the Gaussian-smoothed image $ff(x, y, \sigma)$ at the scale σ . $m(x, y)$ as gradient magnitude and $\theta(x, y)$ as gradient orientation.

$$m(x, y) = \sqrt{|ff(x+1, y, \sigma) - ff(x, y, \sigma)|^2 + |ff(x, y+1, \sigma) - ff(x, y, \sigma)|^2}$$

$$\theta(x, y) = \arctan\left(\frac{ff(x+1, y, \sigma) - ff(x, y, \sigma)}{ff(x, y+1, \sigma) - ff(x, y, \sigma)}\right)$$

In the end, for each extrema point, divide its surrounded 16x16 neighbor points into 4x4 cells. Each cell has 4x4 points. For every 16 cells, 8 bin histogram is calculated from $m(x, y)$ weighted $\theta(x, y)$ at 16 pixels in the cell. The result is a 128 elements descriptor.

The corresponding points are found through Euclidean distance. For Euclidean method, is simply equal to \sqrt{SSD} mathematically.

$$SSD = \sum_i \sum_j |f_1(i, j) - f_2(i, j)|^2$$

$f_1(i, j)$ represents the 128 element descriptor in the first image obtained with coordinate i and j , $f_2(i, j)$ denotes the same meaning. To minimize the result amount in EUC, we take the following two steps:

1. Set a threshold to remove weak candidates, any EUC result has a value less than the threshold will be removed from the final match pair list.
2. If $\frac{\text{minimum of EUC}}{\text{second minimum of EUC}}$ is less than a certain ratio, the result will be dumped from the final match pair list.

Parameter setting:

$5 * \text{absolute value of minimum in EUC matrix}$ -- threshold used to constrain EUC candidates

0.85 -- ratio used constrain EUC

Step 2. RANSAC (Random sample consensus)

Parameter setting:

δ – distance threshold, if the distance is smaller than threshold, then it is an inlier. Otherwise, it is a outlier.

ε – probability that any single correspondence is a false inlier, usually set to 0.1.

p – probability that at least one of the N trials will be free of outliers, set to 0.99.

n – the number of correspondences chosen at each trial.

N – number of iteration of RANSAC, calculate as $\frac{\ln(1-p)}{\ln(1-(1-\varepsilon)^n)}$.

M – threshold for the size of inlier. A inlier has to contain more than M points to be accepted. Calculated as $\text{total number of correspondences} * (1 - \varepsilon)$.

In each iteration, the algorithm started by choosing n number of correspondence from SIFT result. Using **Least Square method** to solve the equation $A\mathbf{h} = \mathbf{0}$, then \mathbf{h} is the calculated homograph.

After getting the homograph, use it to generate the true matching points. Calculate the distance between the true matching points and the SIFT matching points. If the distance is less than the δ , distance threshold, it is considered as a

inlier. Otherwise, it is an outlier.

Finally, after N iteration, compare all the inlier result from each iteration. Select the inlier result with the most points(at least large than M) to be the final result. The corresponding homograph will also be stored.

Step 3. Least square method

Solving the equation of: $\mathbf{Ah} = \mathbf{0}$, the basic format for a single pair (x, x') looks like:

$$\begin{bmatrix} 0 & 0 & 0 & -w'x & -w'y & -w'w & y'x & y'y & y'w \\ w'x & w'y & w'w & 0 & 0 & 0 & -x'x & -x'y & -x'w \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

Where (x, x') represents a matching pair, using n points to build the matrix A in the above format to solve the matrix of homograph.

In this case, we seek for the solution \mathbf{h} which minimize $\|\mathbf{Ah}\|$ with a constraint $\|\mathbf{h}\| = 1$. Therefore, the solution becomes the eigenvector of $\mathbf{A}^T\mathbf{A}$ which corresponds to its smallest eigenvalue.

Step 4. Dog-leg algorithm

Dog-leg algorithm is used to refine homograph. The homograph can be

represent as $\mathbf{P}_k = [\mathbf{h}_{11} \quad \mathbf{h}_{12} \quad \mathbf{h}_{13} \quad \mathbf{h}_{21} \quad \mathbf{h}_{22} \quad \mathbf{h}_{23} \quad \mathbf{h}_{31} \quad \mathbf{h}_{32} \quad \mathbf{h}_{33}]^T$.

The algorithm tends update \mathbf{P}_k to minimize error.

First, Jacobian and error vector need to be computed as:

$$\mathbf{J}(\mathbf{P}_k) = \begin{bmatrix} \frac{\mathbf{x}_1}{\widehat{\mathbf{w}}_1} & \frac{\mathbf{y}_1}{\widehat{\mathbf{w}}_1} & \frac{1}{\widehat{\mathbf{w}}_1} & 0 & 0 & 0 & -\frac{\widehat{\mathbf{x}}_1 \mathbf{x}_1}{\widehat{\mathbf{w}}_1^2} & -\frac{\widehat{\mathbf{x}}_1 \mathbf{y}_1}{\widehat{\mathbf{w}}_1^2} & -\frac{\widehat{\mathbf{x}}_1}{\widehat{\mathbf{w}}_1^2} \\ 0 & 0 & 0 & \frac{\mathbf{x}_1}{\widehat{\mathbf{w}}_1} & \frac{\mathbf{y}_1}{\widehat{\mathbf{w}}_1} & \frac{1}{\widehat{\mathbf{w}}_1} & -\frac{\widehat{\mathbf{y}}_1 \mathbf{x}_1}{\widehat{\mathbf{w}}_1^2} & -\frac{\widehat{\mathbf{y}}_1 \mathbf{y}_1}{\widehat{\mathbf{w}}_1^2} & -\frac{\widehat{\mathbf{y}}_1}{\widehat{\mathbf{w}}_1^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\mathbf{x}_n}{\widehat{\mathbf{w}}_n} & \frac{\mathbf{y}_n}{\widehat{\mathbf{w}}_n} & \frac{1}{\widehat{\mathbf{w}}_n} & 0 & 0 & 0 & -\frac{\widehat{\mathbf{x}}_n \mathbf{x}_n}{\widehat{\mathbf{w}}_n^2} & -\frac{\widehat{\mathbf{x}}_n \mathbf{y}_n}{\widehat{\mathbf{w}}_n^2} & -\frac{\widehat{\mathbf{x}}_n}{\widehat{\mathbf{w}}_n^2} \\ 0 & 0 & 0 & \frac{\mathbf{x}_n}{\widehat{\mathbf{w}}_n} & \frac{\mathbf{y}_n}{\widehat{\mathbf{w}}_n} & \frac{1}{\widehat{\mathbf{w}}_n} & -\frac{\widehat{\mathbf{y}}_n \mathbf{x}_n}{\widehat{\mathbf{w}}_n^2} & -\frac{\widehat{\mathbf{y}}_n \mathbf{y}_n}{\widehat{\mathbf{w}}_n^2} & -\frac{\widehat{\mathbf{y}}_n}{\widehat{\mathbf{w}}_n^2} \end{bmatrix}$$

$$\mathbf{E}(\mathbf{P}_k) = \left[\left(\mathbf{x}_1' - \frac{\widehat{\mathbf{x}}_1}{\widehat{\mathbf{w}}_1} \right) \quad \left(\mathbf{y}_1' - \frac{\widehat{\mathbf{y}}_1}{\widehat{\mathbf{w}}_1} \right) \quad \dots \quad \left(\mathbf{x}_n' - \frac{\widehat{\mathbf{x}}_n}{\widehat{\mathbf{w}}_n} \right) \quad \left(\mathbf{y}_n' - \frac{\widehat{\mathbf{y}}_n}{\widehat{\mathbf{w}}_n} \right) \right]^T$$

Where $(\mathbf{x}, \mathbf{x}')$ represents a matching pair, and $\widehat{\mathbf{x}}_k = \mathbf{h}_{11}\mathbf{x}_k + \mathbf{h}_{12}\mathbf{y}_k + \mathbf{h}_{13}$,
 $\widehat{\mathbf{y}}_k = \mathbf{h}_{21}\mathbf{x}_k + \mathbf{h}_{22}\mathbf{y}_k + \mathbf{h}_{23}$, $\widehat{\mathbf{w}}_k = \mathbf{h}_{31}\mathbf{x}_k + \mathbf{h}_{32}\mathbf{y}_k + \mathbf{h}_{33}$.

Then we compute two parameters to update:

$$\delta_{GD}(\mathbf{P}_k) = \frac{\|\mathbf{J}(\mathbf{P}_k)^T \mathbf{E}(\mathbf{P}_k)\|}{\|\mathbf{J}(\mathbf{P}_k) \mathbf{J}(\mathbf{P}_k)^T \mathbf{E}(\mathbf{P}_k)\|} \mathbf{J}(\mathbf{P}_k)^T \mathbf{E}(\mathbf{P}_k)$$

$$\delta_{GN}(\mathbf{P}_k) = (\mathbf{J}(\mathbf{P}_k)^T \mathbf{J}(\mathbf{P}_k) + \mathbf{u}_k)^{-1} \mathbf{J}(\mathbf{P}_k)^T \mathbf{E}(\mathbf{P}_k), \text{ where } \mathbf{u}_k \text{ is a constant.}$$

From here, we can update \mathbf{P}_k by the following options:

(1). If $\|\delta_{GN}(\mathbf{P}_k)\| < \mathbf{r}_k$

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \delta_{GN}(\mathbf{P}_k)$$

(2). If $\|\delta_{GD}(\mathbf{P}_k)\| < \mathbf{r}_k < \|\delta_{GN}(\mathbf{P}_k)\|$

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \delta_{GD}(\mathbf{P}_k) + \beta(\delta_{GN}(\mathbf{P}_k) - \delta_{GD}(\mathbf{P}_k)), \text{ where } \beta \text{ is the solution of}$$

$$\|\delta_{GD}(\mathbf{P}_k) + \beta(\delta_{GN}(\mathbf{P}_k) - \delta_{GD}(\mathbf{P}_k))\|^2 = \mathbf{r}_k^2$$

(3). Otherwise

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \frac{\mathbf{r}_k}{\|\delta_{GD}(\mathbf{P}_k)\|} * \delta_{GD}(\mathbf{P}_k)$$

Then update \mathbf{r}_k as:

(1) If $\rho\mathbf{DL} < 0.25$

$$\mathbf{r}_{k+1} = \frac{\mathbf{r}_k}{4}$$

(2) If $0.25 \leq \rho\mathbf{DL} \leq 0.75$

$$\mathbf{r}_{k+1} = \mathbf{r}_k$$

(3) Otherwise

$$\mathbf{r}_{k+1} = 2\mathbf{r}_k$$

Then $C(\mathbf{P}_k) = \mathbf{E}(\mathbf{P}_k)^T \mathbf{E}(\mathbf{P}_k)$, also by updating \mathbf{P}_k and re-compute $C(\mathbf{P}_{k+1}) = \mathbf{E}(\mathbf{P}_{k+1})^T \mathbf{E}(\mathbf{P}_{k+1})$, $\delta_k = \mathbf{P}_{k+1} - \mathbf{P}_k$.

The stop condition is $\rho\mathbf{DL} = \frac{C(\mathbf{P}_k) - C(\mathbf{P}_{k+1})}{2\delta_k^T \mathbf{J}(\mathbf{P}_k)^T \mathbf{E}(\mathbf{P}_k) - \delta_k^T \mathbf{J}(\mathbf{P}_k)^T \mathbf{J}(\mathbf{P}_k) \delta_k}$.

Repeat the above algorithm until $\rho\mathbf{DL} < 0$.

Step 5. Stitch image together

In my case, I have 5 images start from the left-view of a building to right-view. Therefore, the stitch procedure is done by mapping the 1st, 2nd, 4th, 5th image onto the 3rd image.

The homograph has already been found by RANSAC as H12,H23,H43,and H54.

We still need to calculate :

$H_{13} = H_{12} * H_{23}$ and $H_{53} = H_{54} * H_{43}$.

Then, by using the homograph method the same as homework 2, we can map all five images on to one image.

Result demo:

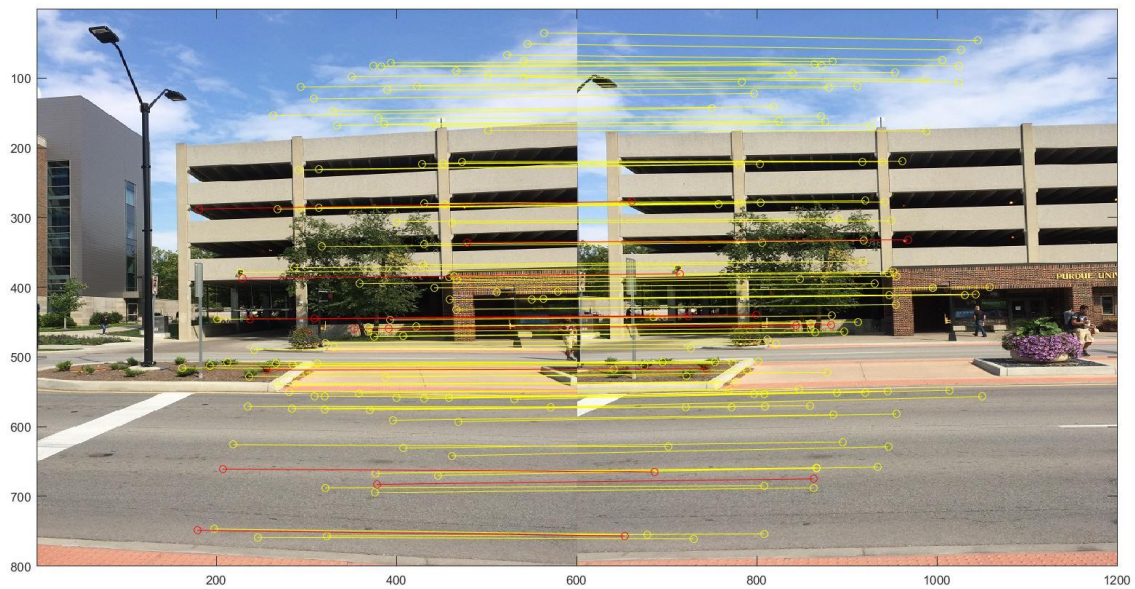


Figure 1. match result between image 1 and image 2, yellow lines represent inliers, red lines represent outliers.

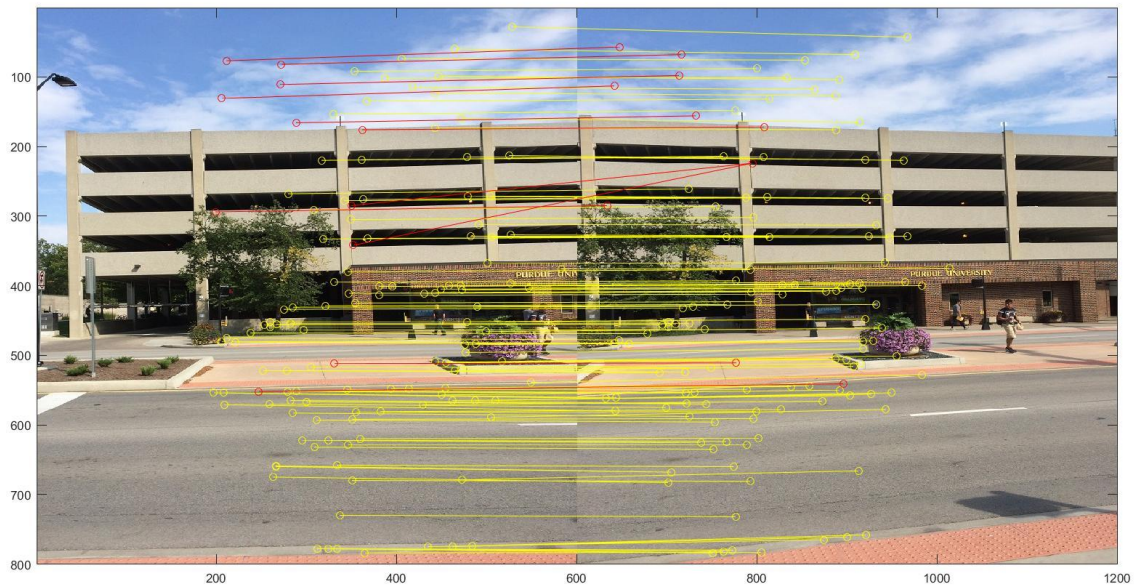


Figure 2. match result between image 2 and image 3, yellow lines represent inliers, red lines represent outliers.

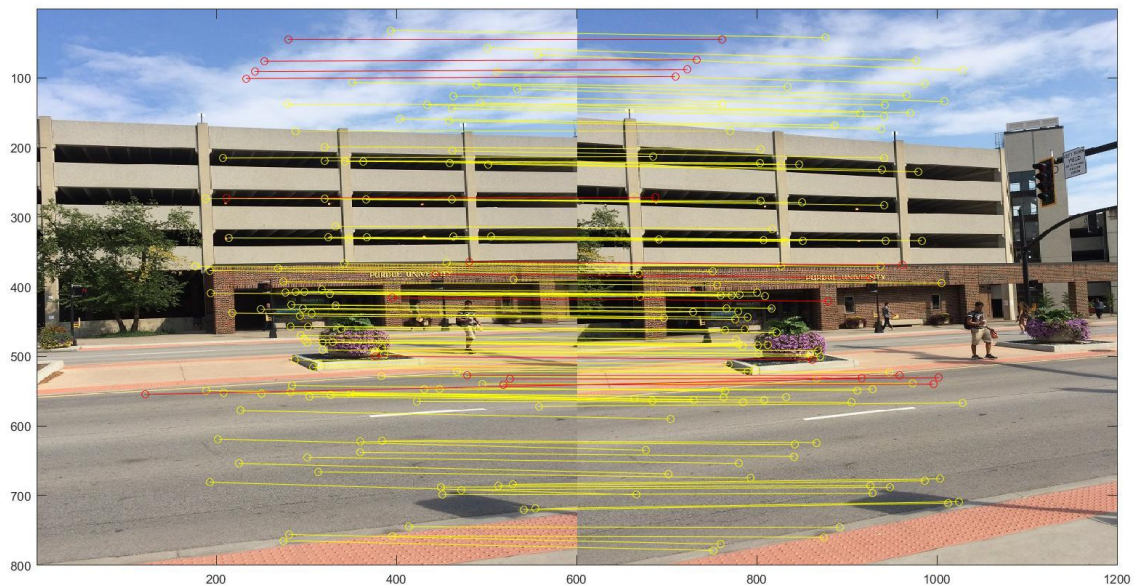


Figure 3. match result between image 3 and image 4, yellow lines represent inliers, red lines represent outliers.

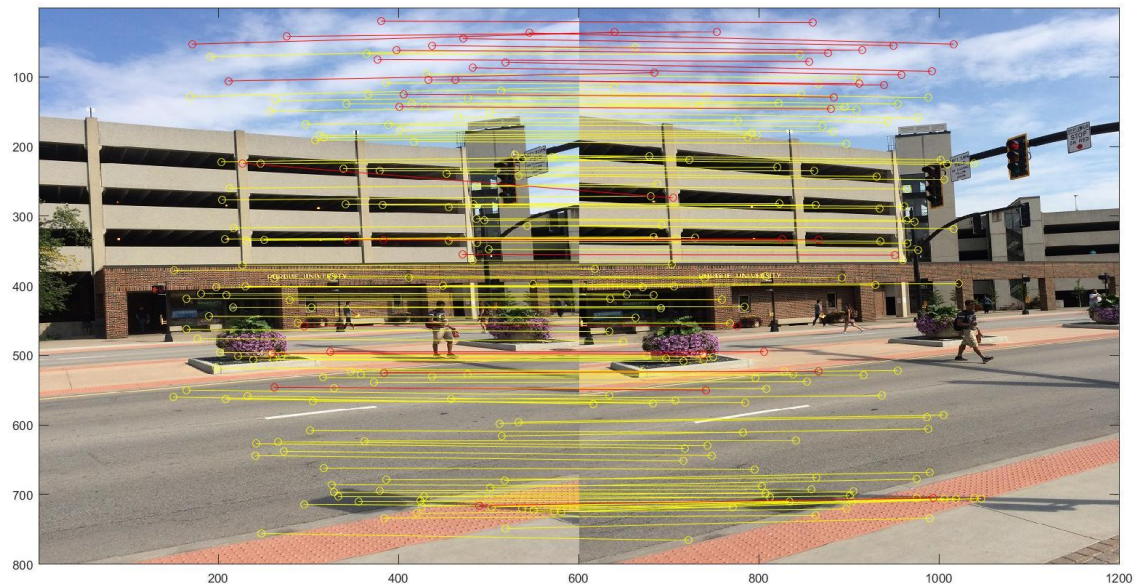


Figure 4. match result between image 4 and image 5, yellow lines represent inliers, red lines represent outliers.



Figure 5. Original images



Figure 6. Image stitch result without Dog-leg algorithm



Figure 7. Image stitch result with Dog-leg algorithm

Conclusion:

- The SIFT algorithm has already done a good job on finding match pairs. RANSAC helped to distinguish inliers and outliers, and we have much more inliers points than outliers.
- By applying Dog-leg algorithm to refine homograph, the resulted image stitch has been improved. Especially on the connection between image 1 and 2, the gap disappears by applying this refine algorithm.