Image Stitching

Group6

December 17, 2023

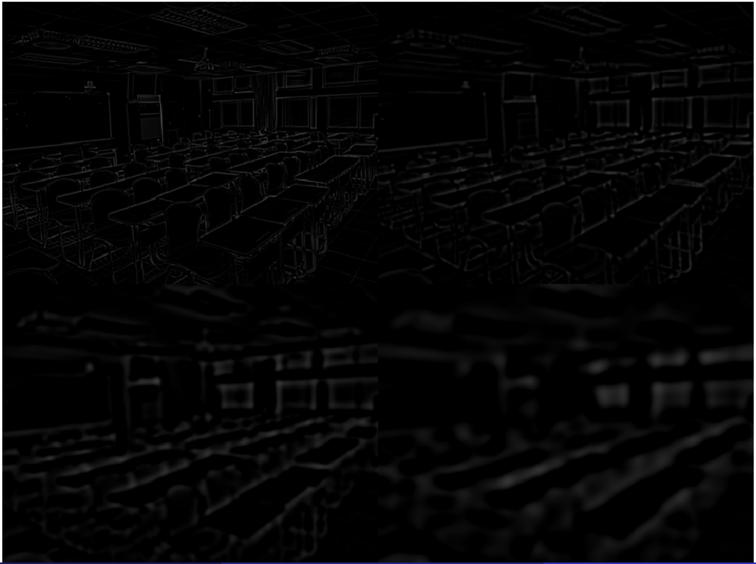
Overview

1. Some challenges in stitching images
2. SIFT
3. SURF
4. Key Point Matching
5. RANSAC
6. Mathematical formulations of image transformations and images wrapping
7. RANSAC in Image Stitching
8. Image blending

Some challenges in stitching images

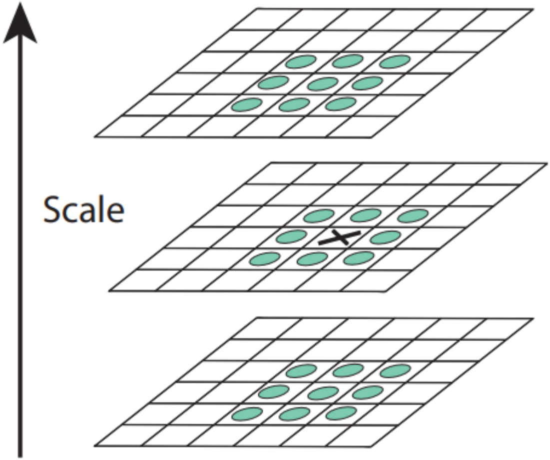
1. How to find the similar points between two images?
2. Calculate the transformation of these matching points.
3. How to eliminate the mistake in matching wrong point by SIFT or SURF?
4. The absorbance of the light in two image might not be totally same.

Difference of Gaussian



SIFT

Location of feature point



SIFT

SIFT

Orientation assignment

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{1}$$

$$\theta(x,y) = tan^{-1}(((L(x+1,y) - L(x-1,y))^2)/((L(x,y+1) - L(x,y-1))^2)) \tag{2}$$

SIFT

Feature Point Descriptor

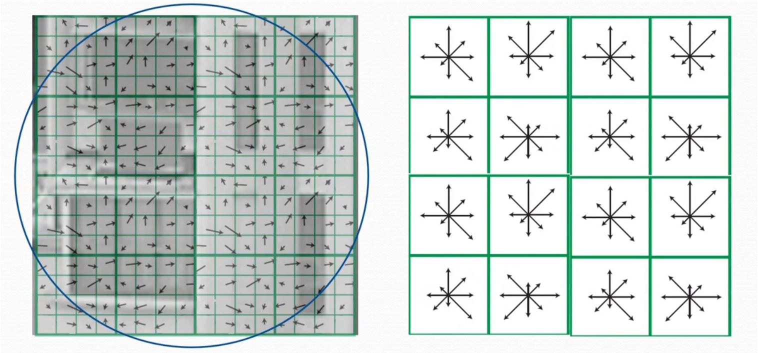


Figure: feature point descriptor

Hessian matrix:

$$H(x, y, \sigma) = \begin{bmatrix} \frac{\partial^2 f((x,y,\sigma))}{\partial x^2} & \frac{\partial^2 f((x,y,\sigma))}{\partial x \partial y} \\ \frac{\partial^2 f((x,y,\sigma))}{\partial x \partial y} & \frac{\partial^2 f((x,y,\sigma))}{\partial y^2} \end{bmatrix} \quad (3)$$

, where selecting points near (x, y) to give a approximate function by Taylor polynomial, saying f.

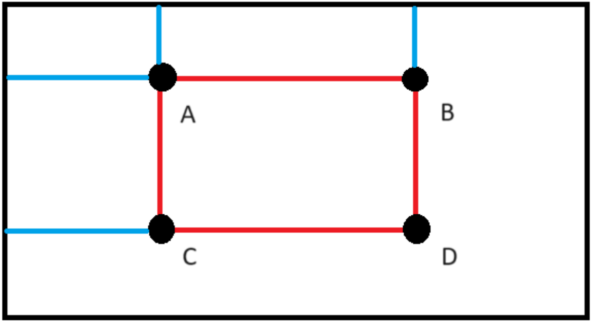
SURF

Approximation of Hessian matrix:

$$Det(H_{approx}) = D_{xx} * D_{yy} - (0.9 * D_{xy})^2 \tag{4}$$

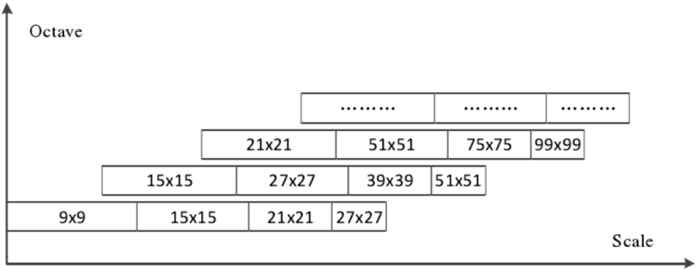
, where D is a function obtained by box filter. In this equation, 0.9 is the deviation with box filter comparing to Gaussian filter.

Integral image:



Sum = D + A - B - C

Scale-space representation:

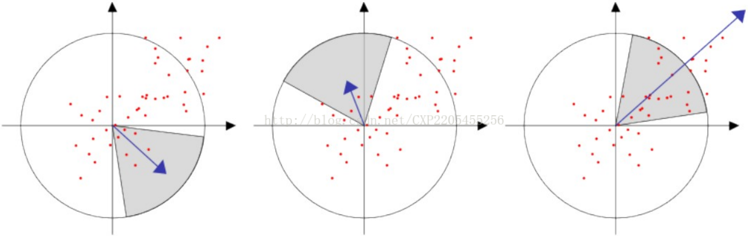


SURF

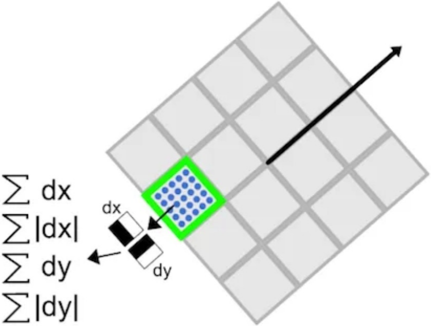
SURF

Orientation assignment:
In SURF algorithm, we first calculate the Haar-wavelet responses in x and y-direction, and this in a circular neighborhood of radius 6s around the feature point, with s the scale at which the feature point was detected. Also, the sampling step is scale dependent and chosen to be s, and the wavelet responses are computed at that current scale s.

SURF



Descriptor representation:



Key Point Matching

The Key Point Matching is designed for finding 1-1 matching for feature points and discarding points with no correct match:

- First, we flatten the data from (n,16,8) to the size (n,128) , and define distance of two keypoints by manhattan distance $d(x, y) = \sum_{j=1}^{m} |a_{ij} - b_{ij}|$ for boosting the calculation.
- Second, we fix a point, and iterate all key points in second picture to find the keypoint match that minimize the distance(p1,p2). we repeat the process for all point in first picture, then finding all (number of n) matches.

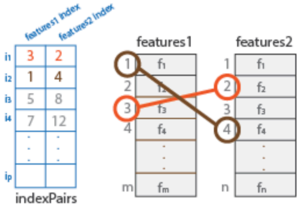


Figure: a quick illustration of keypoint matching.

Key Point Matching

Problem : many features from an image would not have a correct match due to background clutter or were not detected in the training images.
Solution : We define ratio of distances by closest distance divided by second-closet distance. By eliminating matches with ratio of distances larger than 0.8, we reduce 90% of false matches with only 5% of correct matches discard.

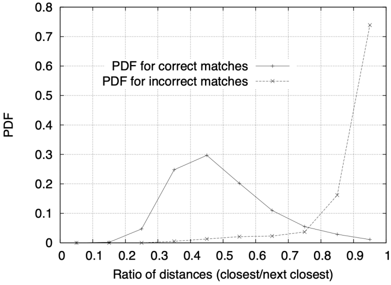


Figure: pdf of correct and incorrect matches

RANSAC

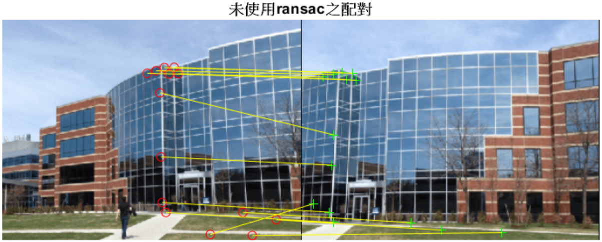


Figure:

RANSAC

Definition of inliers and outliers:
Inliers:The data points that are considered consistent with the suitable model being estimated.
Outliers:The data points that deviate significantly from the suitable model.

RANSAC

The RANSAC algorithm is composed of these steps:

- Choose s samples "randomly". s is the minimum samples to fit a model
- Find the suitable model to the randomly chosen samples
- Count the number of data points (inliers) that fit the model within a measure of error e
- Repeat the following steps N times
- Choose the model generated by this N times. Find the model that has the largest number of inliers

RANSAC

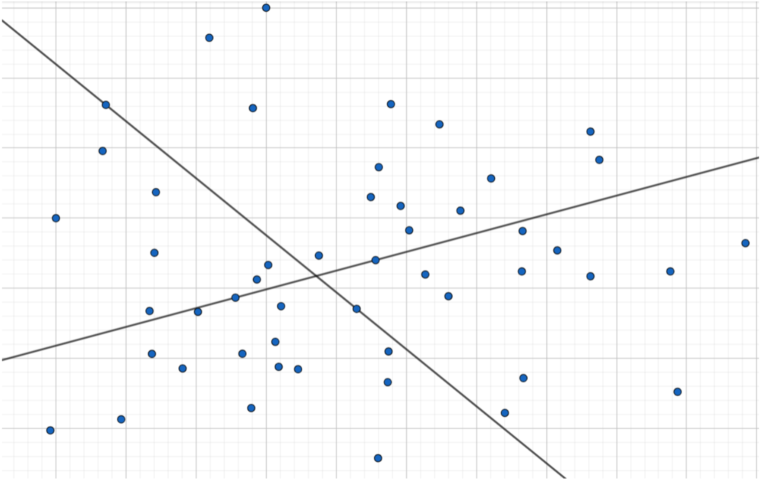
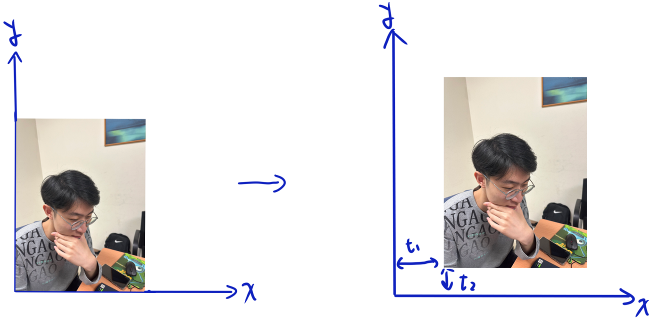


Figure:

Mathematical formulations of image transformations and images wrapping

How to transform figure A to figure B?



Mathematical formulations of image transformations and images wrapping

I mean a transformation T satisfied

$$\begin{cases} x_2 = x_1 + t_1 \\ y_2 = y_1 + t_2 \end{cases} \tag{5}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + t_1 \\ y_1 + t_2 \end{bmatrix} = T \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{6}$$

Mathematical formulations of image transformations and images wrapping

identity

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

reflection

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

scaling

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

rotation

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos & \sin \\ -\sin & \cos \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

shearing

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Mathematical formulations of image transformations and images wrapping

Figure: identity

Figure: reflection

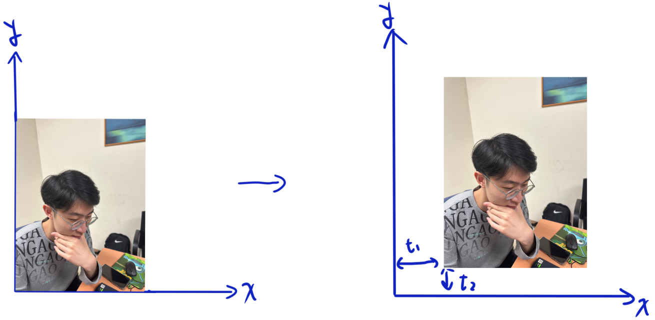
Figure: scaling

Figure: rotation

Figure: shearing

Mathematical formulations of image transformations and images wrapping

How to transform figure A to figure B?



Mathematical formulations of image transformations and images wrapping

we have to use homogeneous coordinate to show translation of image. That means for any point (x, y) in $\mathbb{R}^2$, $(x\mathbb{Z}, y\mathbb{Z}, \mathbb{Z})$ is its homogeneous coordinate. For example, point $(1, 2)$ can be writen as $(1, 2, 1)$ or $(2, 4, 2)$ in homogeneous coordinate. So, the transformation of translate is

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{7}$$

Affine transformation includes the below transformation we talked about(rotation, shearing, translation...), and it can be writen as

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{8}$$

Mathematical formulations of image transformations and images wrapping

Now, we know about homogeneous coordinate. Then we consider

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ z_1' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{9}$$

This is called homography transformation. We usually let $\|H\|^2 = 1$

Mathematical formulations of image transformations and images wrapping

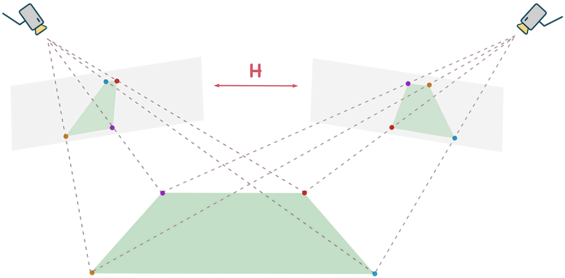


Figure: homography transformation

Mathematical formulations of image transformations and images wrapping

To compute homography matrix H, we rewrite the equation into

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_2x_1 & -x_2y_1 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_2x_1 & -y_2y_1 & y_2 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{10}$$

Mathematical formulations of image transformations and images wrapping

Then solve the h_{ij} by using the characteristic points pairs,

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_2x_1 & -x_2y_1 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_2x_1 & -y_2y_1 & y_2 \\ x_1^{(2)} & y_1^{(2)} & 1 & 0 & 0 & 0 & -x_2^{(2)}x_1^{(2)} & -x_2^{(2)}y_1^{(2)} & -x_2^{(2)} \\ 0 & 0 & 0 & x_1^{(2)} & y_1^{(2)} & 1 & -y_2^{(2)}x_1^{(2)} & -y_2^{(2)}y_1^{(2)} & y_2^{(2)} \\ x_1^{(3)} & y_1^{(3)} & 1 & 0 & 0 & 0 & -x_2^{(3)}x_1^{(3)} & -x_2^{(3)}y_1^{(3)} & -x_2^{(3)} \\ 0 & 0 & 0 & x_1^{(3)} & y_1^{(3)} & 1 & -y_2^{(3)}x_1^{(3)} & -y_2^{(3)}y_1^{(3)} & y_2^{(3)} \\ \dots \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \dots \end{pmatrix} \tag{11}$$

Mathematical formulations of image transformations and images wrapping

Now, we simplify it as $Ah = 0$ with $\|h\|^2 = 1$. To solve this question, we use the method called constrained least square. Change the question into solving

$$\min_h \|Ah\|^2 \tag{12}$$

such that $\|h\|^2 = 1$.
And we can rewrite it as

$$\min_h h^T A^T A h \tag{13}$$

such that $h^T h = 1$.

Mathematical formulations of image transformations and images wrapping

Define Loss function

$$L(h, \lambda) = h^T A^T A h - \lambda(h^T h - 1) \tag{14}$$

and we want to find h that minimized L.
Thus, we compute the differential of L w.r.t. h, and it becomes

$$2A^T A h - 2\lambda h = 0 \tag{15}$$

Rewrite the equation as

$$A^T A h = \lambda h \tag{16}$$

We find the eigenvector h which has the smallest eigenvalue λ. The h is what we looking for.
Actually, we can use only 4 pairs of matching points to find this homography matrix.

RANSAC in Image Stitching

未使用ransac之配對

RANSAC in Image Stitching

1. choose 4 pairs of points we match before, then we can construct a homogeneous matrix H_1.
2. put all the characteristic points in image1 into the transformation and get the new points.
3. calculate the difference between the new points and the points we pair to the original characteristic points. If it is smaller than a constant ϵ we choose, we called this pair of points "inliers".
4. Repeated these steps n times(In our code, we run 1000 times), then we can get $H_1, H_2, ..., H_n$ and the number of inliers every transformation has.
5. we choose the matrix which have the most inliers to be the transformation between image1 and image2.

RANSAC in Image Stitching

使用ransac後匹配的內點

Image blending

Figure: without modify

Image blending

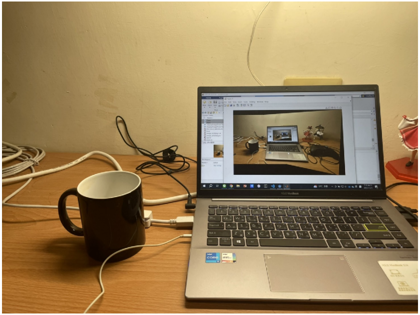


Figure: photo1

Figure: photo2

Image blending

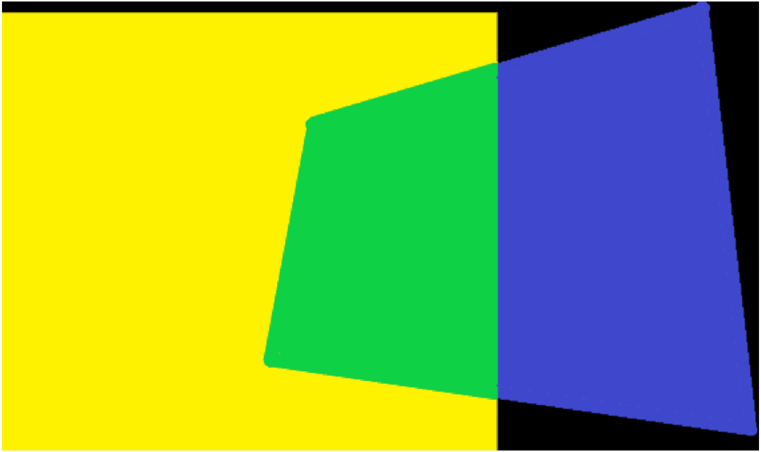


Image blending

bwdist()

```
a =

     0     1     0     0     0
     0     0     0     0     0
     0     0     0     1     0
     0     0     0     0     0
     0     0     0     0     0
```

```
1.0000         0    1.0000    2.0000    2.2361
1.4142    1.0000    1.4142    1.0000    1.4142
2.2361    2.0000    1.0000         0    1.0000
3.1623    2.2361    1.4142    1.0000    1.4142
3.6056    2.8284    2.2361    2.0000    2.2361
```

Image blending

Figure: after weighted

Image blending

Figure: photos with large difference of exposure value

Image blending

```
sum_1 = sum(I1_gray, "all");
sum_2 = sum(I2_gray, "all");
exposure_value1 = sum_1 / (size(I1,1)*size(I1,2));
exposure_value2 = sum_2 / (size(I2,1)*size(I2,2));
mean_expo_val = (exposure_value1+exposure_value2) / 2;

I1 = I1 * (mean_expo_val/exposure_value1);
I2 = I2 * (mean_expo_val/exposure_value2);
```

Image blending

Figure: after modify the lightness