DigiVFX HW2 - Image Stitching

MINHSUAN CHENG / P10942A05

April 27, 2022

1 Abstract

Following the method outlined in the course, we will use SIFT to extract image features. The Panoramas pipeline is used to implement feature detection, feature matching, and image stitching using a blending mechanism. You can insert a sequence of images into an image file and execute our code to get the image stitching output. We also allow you to adjust several parameters to improve the quality of your images. The outcome will be saved as "result.png."

2 Implement

2.1 Cylinder Projection

First we use cylinder projection to transform our raw images. In our implementation we need to set up focal length and get corresponding cylinder projection result.

2.2 Scale-Invariant Feature Transform

We use SIFT [Low04] as our key points detection and description. [Isl21]

2.2.1 Get Gaussian Images

Follow paper description we set octave as 4 and number of interval as 2. We can get 4 octaves and per octave have 5 images, total 20 images. We use different Gaussian blur intensity to apply images in octave.First we set first image Gaussian blur intensity as sigma. Calculate intensity difference in per images and set intensity 2 Gaussian image as next octave base image Figure 1 . At last we prepare all Gaussian image to next step operation.

$$OctaveImageNumbers = NumberOfIntervals + 3$$

$$GaussianBlurDifference = 2^{\frac{1}{NumberIntervals}}$$

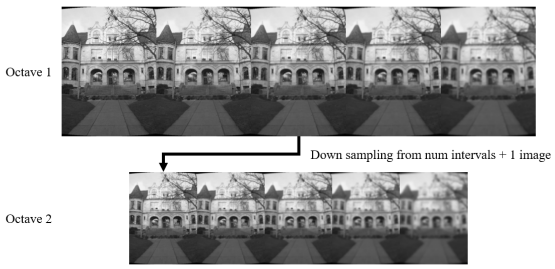

Figure 1: Denny image stitching result without crop

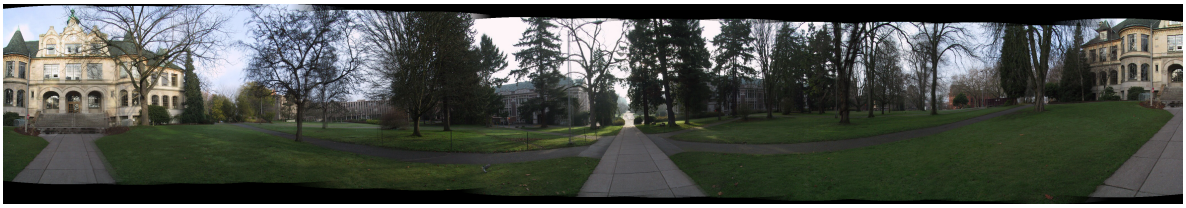

Figure 2: Denny image stitching result without crop

Figure 3: Denny image stitching result crop

2.2.2 Calculate DOG

We calculate per octave difference of Gaussian(DOG) and select the maximum difference points as our feature point using preprocessing photos.

2.2.3 Find extreme points and calculate orientation

We apply a quadratic function to calculate real extreme value position to fine-tune our feature point location. We'll also eliminate several points with extreme values that aren't in the center. The next step is to determine the orientation of the feature point. To calculate per bin intensity in different places, we utilize 36 bins in 360 degrees (each bin represents a 10 degree zone). Find the bin's maximum intensity and use 0.8 multiply max bin intensity as a threshold to determine whether we need to orient. Assign an orient value to the key point angle data. Finding key points has come to an end.

2.2.4 Calculate descriptors

We compute key point descriptions using SIFT algorithms, which include 8 bins in 16 neighboring regions for a total of 128 dimension information. We first restore picture information from key points, then change the scale size of a selected region using the scale size of a key point located region. Then, from the specified location, we calculate 128 dimension information. We must use trilinear interpolation to obtain accurate results. Calculate descriptors is now complete.

2.3 Brutal Force Match

To match image key points we use brutal force match method. Calculate two image key points euclidean distance as length(image 1 key points) X length(image 2 key points) dimension matrix. We find the minimum distance in per row and check threshold multiply second minimum value whether bigger than minimum distance. If meet above condition we select key points as our good matches result.

2.4 Get Image Shift

In this function we select RANSAC method to random select points and calculate its shift values (x shift and y shift). Then calculate total error in all matched key points use above shift result. At last we select minimum error as our shift value.

2.5 Image Stitching with blending

Last step of our implementation is stitching image. To get better stitching image result we use linear interpolation blending in image cross regions. Figure 2 Next use image shift result to fine tune image location and crop image to reduce image shift impact. Figure 3

Figure 4: Our image stitching result use above parameters set up.

3 Result

In our implementation, we resize image height and width to 1/3 raw image size to reduce SIFT calculate time. Cylinder projection we set focal length as 1600. Set contrast threshold 0.1 and number of intervals 2, initial sigma 1.6 and assumed blur 0.5. Set higher contrast threshold can reduce our detect numbers of key point, then reduce computation time. Brutal force match method we set 0.6 as our threshold. Get image shift RANSAC we set 12 as our per sampling key points and 100 times iteration.Figure 4

4 Conclusion

In this implementation SIFT will use most of computation time. So we modify some parameters and reduce image size to accelerate our process. We think next time can use SURF or ORB methods to detect image key points to accelerate our implementation time. Or parallel compute SIFT to accelerate. Image blending method use simple linear interpolation can modify as gradient based method to get better result.

References

[Isl21] Russ Islam. A clean and concise python implementation of sift (scale-invariant feature transform). https://github.com/rmislam/PythonSIFT, 2021.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004.