

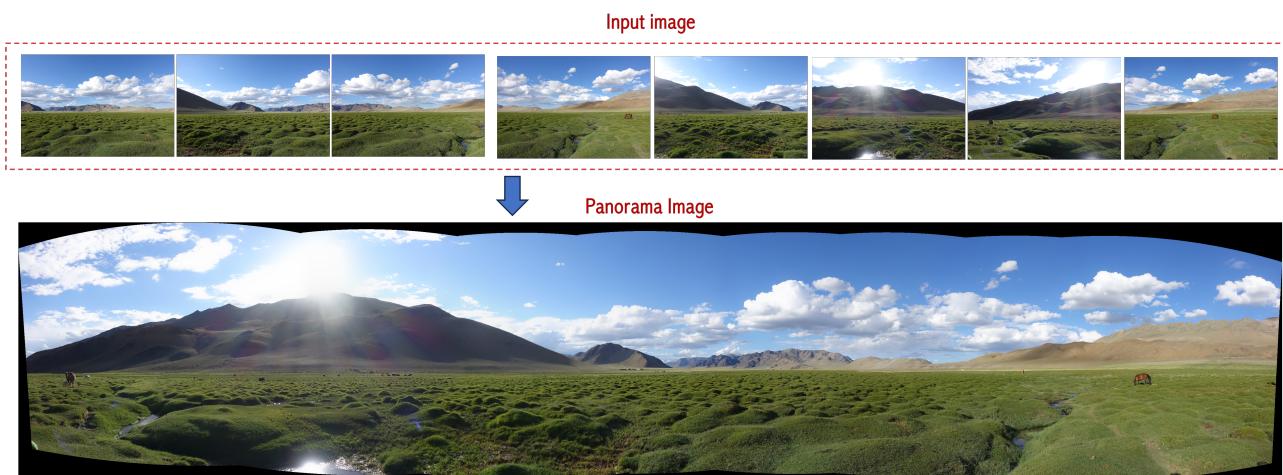
AI VIET NAM – COURSE 2023

Panorama Image: Stitching Technique (Computer Vision Foundation)

Ngày 12 tháng 10 năm 2023

Giới thiệu về bài tập sử dụng kỹ thuật image stitching để xây dựng ứng dụng ảnh thông dụng Panorama:

Chức năng chụp ảnh panorama của Smart phone khá thông dụng hiện nay. Chúng ta có thể tạo ra ảnh panorama bằng cách lia camera từ từ qua khung cảnh cần chụp. Ở bài tập này các bạn sẽ được làm quen với kỹ thuật Image Stitch, đây là công nghệ đầu tiên sau ứng dụng panorama. Đề bài của bài tập này là cho trước một bộ 8 hình ảnh được chụp rời rạc từ camera, các bạn hãy sử dụng kỹ thuật Image Stitching để tạo ra ảnh panorama từ 8 ảnh này. Hình 1, thể hiện kết quả sau khi thực thi giải thuật Image Stitching trên bộ ảnh trên.



Hình 1: Kết quả panorama với kỹ thuật image stitching

Để hoàn thành bài tập này bạn cần từng bước hoàn thiện các bài tập nhỏ bên dưới.

Bài tập 1: (Làm quen với SIFT feature) Hãy hoàn thiện đoạn code sau để tìm ra các đặc trưng (feature) sử dụng giải thuật SIFT bằng thư viện OpenCV. Hình 2 thể hiện kết quả sau khi thực thi SIFT.

```
1 #Question 1
2 import numpy as np
3 import cv2
4 import imageio
5 from google.colab.patches import cv2_imshow
6
7 img = cv2.imread('/content/1.jpg')
8 gray= cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
9
10 sift = # your code here *****
```

```

11 kp = #your code here ****
12 img=cv2.drawKeypoints(img,kp,img, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
13
14 cv2_imshow(img)
15 print("Total keypoints: ", len(kp))

```

Question 1: Hãy cho biết kết quả sau khi thực hiện đoạn code ở bài tập 1

- a) Total keypoints: 8878
- b) Total keypoints: 9878
- c) Total keypoints: 6878
- d) Total keypoints: 7878



Hình 2: Kết quả phát hiện feature bằng giải thuật SIFT

Bài tập 2: (Làm quen với kỹ thuật Transformation) Hãy hoàn thiện đoạn code bên dưới để thực hiện scale và quay ảnh đầu vào 45 độ. Hình 3 thể hiện kết quả trước và sau khi thực hiện các kỹ thuật transformation trên ảnh.

```

1 #Question 2
2 import cv2
3 import matplotlib.pyplot as plt
4 import numpy as np
5 %matplotlib inline
6
7 # Load the image
8 original_img = cv2.imread('/content/1.jpg')
9
10 # Create test image by adding Scale and Rotational on image
11 rotation_image = cv2.pyrDown(original_img)
12 num_rows, num_cols = rotation_image.shape[:2]
13

```

```

14 # your code here ****
15
16 cv2_imshow(original_img)
17
18 cv2_imshow(rotation_image)
19
20 gray= cv2.cvtColor(rotation_image ,cv2.COLOR_BGR2GRAY)
21 sift = cv2.SIFT_create()
22 kp = sift.detect(gray ,None)
23
24 print("Total keypoints of the rotation image:", len(kp))

```

Question 2: Hãy cho biết kết quả sau khi thực hiện đoạn code ở bài tập 2

- a) Total keypoints of the rotation image: 1248
- b) Total keypoints of the rotation image: 2248
- c) Total keypoints of the rotation image: 3248
- d) Total keypoints of the rotation image: 4248



Trước



Sau

Hình 3: Kết quả sau khi thực hiện kỹ thuật transformation trên ảnh

Bài tập 3: (Làm quen với kỹ thuật SIFT matching) Hãy hoàn thiện đoạn code sau để tìm các feature tương đồng giữa ảnh gốc và ảnh sau khi thực hiện transformation. Hình 4 thể hiện kết quả của giải thuật với 5 matching points.

```

1 #question 3
2 orginal_gray = cv2.cvtColor(original_img ,cv2.COLOR_BGR2GRAY)
3 rotation_gray = cv2.cvtColor(rotation_image ,cv2.COLOR_BGR2GRAY)
4
5 sift = cv2.xfeatures2d.SIFT_create()
6 original_keypoints, original_descriptor = sift.detectAndCompute(orginal_gray , None)
7 rotation_keypoints, rotation_descriptor = sift.detectAndCompute(rotation_gray , None)
8
9 # Create a Brute Force Matcher object.
10 bf = cv2.BFMatcher(cv2.NORM_L2)
11
12 # Perform the matching between the SIFT descriptors of the training image and the test
13 # image
14 # your code here ****
15 cv2_imshow(result)
16

```

```

17
18 # Print the best matching feature between the original and rotation images
19 p1 = original_keypoints[matches[0].queryIdx].pt
20 p2 = rotation_keypoints[matches[0].trainIdx].pt
21 print("p1 = ", p1, ", p2 = ", p2)

```

Question 3: Hãy cho biết kết quả sau khi thực hiện đoạn code ở bài tập 3

- a) p1 = (115.92, 360.52), p2 = (95.96, 452.55)
- b) p1 = (114.92, 360.52), p2 = (95.96, 452.55)
- c) p1 = (116.92, 360.52), p2 = (95.96, 452.55)
- d) p1 = (117.92, 360.52), p2 = (95.96, 452.55)



Hình 4: Kết quả sau khi thực hiện kỹ thuật SIFT matching tìm 5 feature tương đồng nhất giữa ảnh gốc và ảnh rotation

Bài tập 4: (Làm quen với kỹ thuật image stitching) Hãy hoàn thiện đoạn code sau để tạo ảnh panorama từ 2 ảnh đầu vào cho trước sử dụng kỹ thuật image stitching. Hình 5 thể hiện kết quả giải thuật image stitching.

```

1 #Question 4
2 src_img = cv2.imread('/content/1.jpg')
3 tar_img = cv2.imread('/content/2.jpg')
4
5 src_gray = cv2.cvtColor(src_img, cv2.COLOR_BGR2GRAY)
6 tar_gray = cv2.cvtColor(tar_img, cv2.COLOR_BGR2GRAY)
7
8 SIFT_detector = cv2.xfeatures2d.SIFT_create()
9 kp1, des1 = SIFT_detector.detectAndCompute(src_gray, None)
10 kp2, des2 = SIFT_detector.detectAndCompute(tar_gray, None)
11
12 # Create a Brute Force Matcher object.
13
14 # your code here *****
15

```

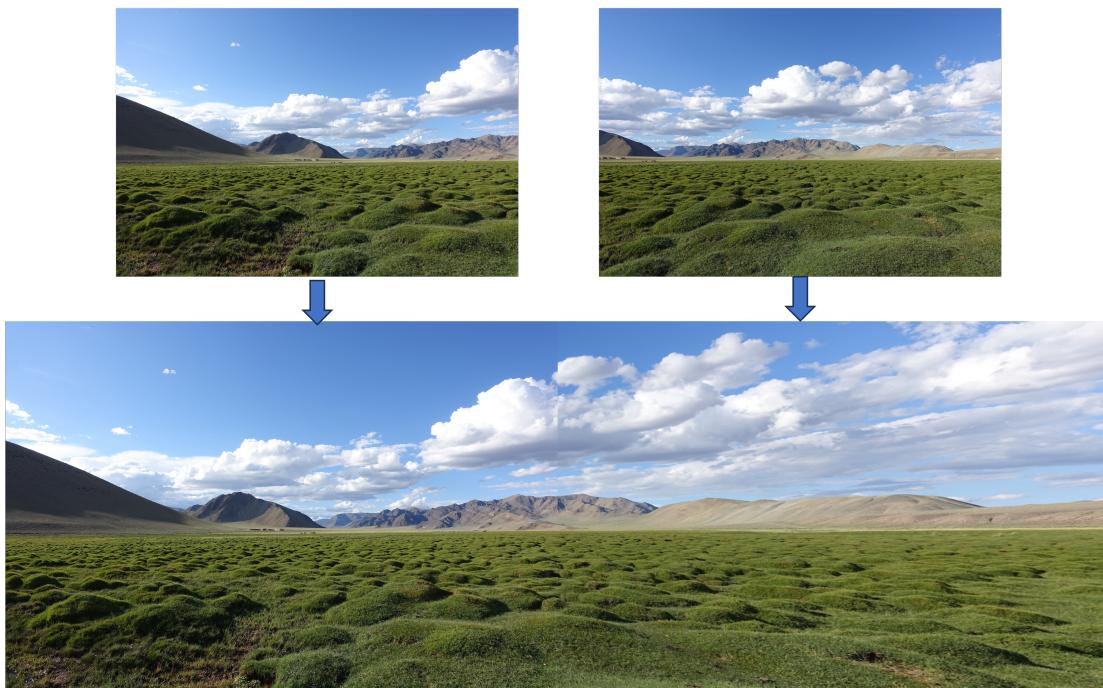
```

16 pts1 = np.float32([kp1[m.queryIdx] for m in matches])
17 pts2 = np.float32([kp2[m.trainIdx] for m in matches])
18
19 # estimate the homography between the sets of points
20 (H, status) = cv2.findHomography(pts1, pts2, cv2.RANSAC)
21 print(H)
22
23 h1, w1 = src_img.shape[:2]
24 h2, w2 = tar_img.shape[:2]
25
26 result = cv2.warpPerspective(src_img, H, (w1+w2, h1))
27 result[0:h2, 0:w2] = tar_img
28 cv2.imshow(result)

```

Question 4: Hãy cho biết kết quả homography sau khi thực hiện đoạn code ở bài tập 4

- a) [[1.58688927e-01 9.51468376e-02 6.76428266e+02]
[-2.39905529e-01 7.84834478e-01 1.27861412e+02]
[-3.86178294e-04 -2.59909552e-05 1.00000000e+00]]
- b) [[2.58688927e-01 9.51468376e-02 6.76428266e+02]
[-2.39905529e-01 7.84834478e-01 1.27861412e+02]
[-3.86178294e-04 -2.59909552e-05 1.00000000e+00]]
- c) [[4.58688927e-01 9.51468376e-02 6.76428266e+02]
[-2.39905529e-01 7.84834478e-01 1.27861412e+02]
[-3.86178294e-04 -2.59909552e-05 1.00000000e+00]]
- d) [[3.58688927e-01 9.51468376e-02 6.76428266e+02]
[-2.39905529e-01 7.84834478e-01 1.27861412e+02]
[-3.86178294e-04 -2.59909552e-05 1.00000000e+00]]



Hình 5: Kết quả sau khi thực hiện kỹ thuật image stitching để tạo ảnh panorama từ 2 ảnh đầu vào

Bài tập 5: (Kỹ thuật image stitching trên nhiều ảnh) Hãy hoàn thiện đoạn code bên dưới để tạo ảnh panorama từ 8 ảnh đầu vào như kết quả ở hình 1 sử dụng thư viện OpenCV.

```

1 imgs = []

```

```
2 for i in range(8):
3     imgs.append(cv2.imread('/content/{}.jpg'.format(i+1)))
4
5 def stitch_images(images):
6     # Create a Stitcher object
7     stitcher = cv2.Stitcher_create()
8
9     # Stitch the images together
10    #your code here ****
11
12    # Return the result
13    return result
14
15 result = stitch_images(imgs)
16 cv2_imshow(result)
```