Panoramic Image Stitching

Siyuan Hong
Department of Electrical and Computer Engineering
New York University
Brooklyn, NY 11201, USA
sh6159@nyu.edu

Haotian Zhu
Department of Electrical and Computer Engineering
New York University
Brooklyn, NY 11201, USA
hz1526@nyu.edu

Yixiang Mao (Mentor)
Department of Electrical and Computer Engineering
New York University
Brooklyn, NY 11201, USA
ym1496@nyu.edu

***Abstract*—Methods of image stitching are well illustrated by Szeliski R [1]. Panoramic image stitching is like an extension of image stitching since it is to register, deform, align, wrap and project a sequence of 360-degree images to a cylinder or spherical surface. Usually images are taken by a single camera, camera arrays or fisheye lens cameras arrays [2]. Panoramic image stitching has several applications, including street view and virtual tours. In this project, we accomplished panoramic image stitching using cylinder projection and have done part job of spherical projection. Source codes, images used and results can be found by this URL: https://github.com/hongzhong558/IVP_Project.**

***Index Terms*—stitching, panoramic, cylinder, spherical**

I. Introduction

Image stitching is an old problem in the area of image processing and computer vision. It is used to solve the problem of limited field of view (FOV). In recent years, it has been applied in our daily lives, e.g., we can take a panoramic photo by setting smartphone camera in the 'PANO' mode. Image stitching takes four steps to form a larger image from original images with feature correspondence. Firstly, the pre-calibration of camera parameters reveals the relationship among original images [3]. Harris [4] or SIFT [5] feature detector is usually applied to detect features, followed by SIFT descriptor for each feature point. Secondly, a plane for projection needs to be confirmed with RANSAC [6]. This can be a plane of an original image or a new plane generated between two images' planes. After that original images are aligned to the projection plane by finding the mapping matrix of corresponding points pairs. Lastly, the part with common features of original images will be blended and remaining part of original images will be preserved. Usually image stitching assumes the imaged scene is planar, but this is appropriately true when the scene is far from the camera and all objects can be seen to share the same depth. Other research about image stitching can be found in [7] and [8]. Although technology of stitching two or three images to a plane have developed for a long time, it can not address the stretching problem when more images are stitched. This problem is shown in Fig. 1.

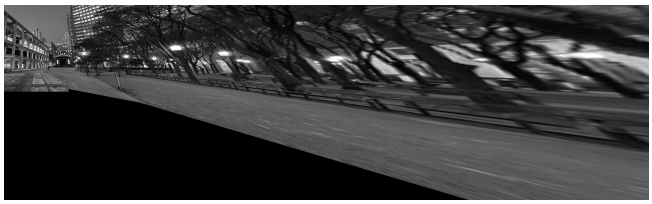

Fig. 1. Stretching Problem

II. Cylindrical Panorama

A. Description

Cylindrical panorama maps a set of images to a surface of cylinder with a radius f, which is the focal length of a camera. The relationship between two coordinates is shown in Fig. 2 [9]. The left part of the picture is the left view and the right part is the top view. Red lines in both parts are the same line, so they share the same length.

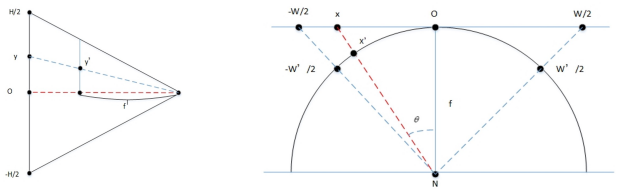

Fig. 2. Principal of Cylindrical Projection

From the top view, we can get:

$$\theta = arctan\frac{x}{f}$$

and

$$x' = f * \theta$$

From the left view we can get:

$$\frac{\frac{y}{f}}{cos(\theta)} = \frac{y'}{f}$$

After applying cylindrical projection, the original rectangle image is twisted. During stitching process pixels on original image is not projected to a plane but a surface of a cylinder. So the stretching problem can be solved. However, cylindrical panorama has more fastidious requirements on acquisition environments. For instance, images need to be taken by rotating a camera 360 degree around its optical flow.

B. Method

Firstly, we use `image_path` to store the directories of all the pictures that need to be stitched. Then we enumerate every directory in `image_path`, and append the picture into a list called `imgs`. After loading every picture, we do the cylindrical projection to `imgs`. Then can be then used as the input to the `stitch` function we defined. In the function we use SIFT as feature detector and kNNMatch to find corresponding point pairs. Then we find homogragphy with RANSAC, followed by aligning and blending images. At last we will have the output, which is the stitched and cropped picture.

The pseudo code is as below:

Input: Pictures need to be stitched
Output: Stitched picture

Store the directory of images in `ImagePath`
Create `imgs` to store the loaded images.
for *image* in *ImagePath* **do**
    Use `cv2` to read image and append to `imgs`.
    Use `cv2` to show image that was loaded.
end for
Do cylindrical projection to `imgs`
Use `stitch` function we defined to stitch pictures in `imgs`
Save the stitched picture into `output`
return *output*

III. Spherical Projection

A. Description

Spherical projects images to a surface of a sphere with a radius r. The transformation between two coordinates is shown in Fig. 3 [10]. The reason of using spherical projection is to assume the original picture is at the surface of a sphere. P is the source of the light and $P2$ is the coordinates of the picture. $P3$ is the projection of $P2$ by P.

According to the definition,

$$P3 = P + k * (P2 - P)$$

and

$$P3 = (px, py, pz)$$

Therefore, we can get:

$$k = (pz - h)/(z - h)$$

$$px = kx$$

$$py = ky$$

After applying the spherical projection, the original picture will be more rounded, since we assume these pictures need to form the surface of a sphere. Therefore, if the pictures are not taken to form a cycle, the result will be comparatively less ideal.

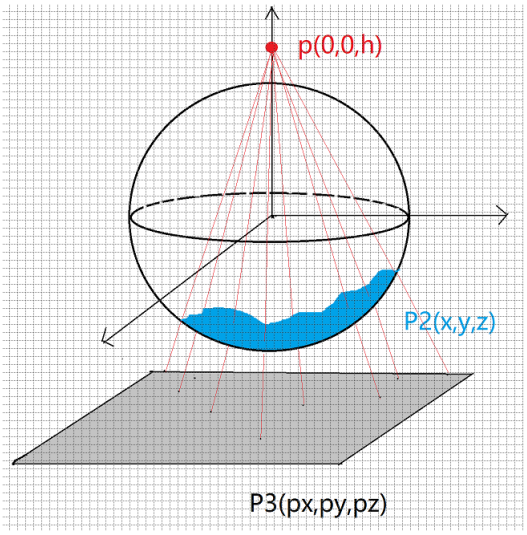

Fig. 3. Principal of Spherical Projection

B. Method

The whole process is similar to cylindrical projection. This section mainly discusses the different steps from cylindrical projection. After loading every picture, we apply spherical projection to these pictures and use the transformed pictures as the input of SIFT and kNNMatch. The following steps are the same as cylindrical projection.

The pseudo code is as below:

Input: Pictures need to be stitched
Output: Stitched picture

Store the directory of images in `ImagePath`
Create `imgs` to store the loaded images.
for *image* in *ImagePath* **do**
    Use `cv2` to read image and append to `imgs`.
    Use `cv2` to show image that was loaded.
end for
Do spherical projection to `imgs`
Use `stitch` function we defined to stitch pictures in `imgs`
Save the stitched picture into `output`
return *output*

IV. Experiments

A. Cylindrical Panorama

1) Test Images: We took several sets of images using iPhone 12 Pro (focal length = 26mm) in different places, e.g., in front of the Bern Dibner Library, 6 MetroTech Center and Roosevelt Island, to take scene depth into account. Moreover, they were taken at different time of a day like afternoon and night to consider the effect of illumination change. More details are shown in the table below.

Location	Depth	Time
Library	Close	Afternoon
6 MetroTech	Close	Night
Roosevelt Island	Far	Afternoon

2) Focal Length in Pixels fc: Setting the radius of cylinder equal to the focal length in pixels fc of the camera is the dirty secret to generate a good cylindrical panorama. The relationship between focal length in millimeter f and focal length in pixels fc is

$$f = fc * \frac{SensorWidth}{ImageWidth}$$

Since we only have the values of f and $ImageWidth$, fc cannot be calculated directly. So we applied binary search algorithm to appropriate the value of fc. After testing on six different values, we found fc is a value around 540.

3) Stitching Results of Library: The result stitched image is shown in Fig. 4. It stitched eight photos together. The center of the image is larger than both edges. There are two reasons for this phenomenon. One is that when taking photos the smartphone was not vertical and the angel of elevation was too large to be neglected. The second reason is that the depth of scenes has a large range. They can hardly be mapped to a homography without twisting.

Fig. 4. Stitching Result of Library

4) Stitching Results of MetroTech: Fig. 5 described the result of stitching two images taken in front of 6 MetroTech in the evening. We can see there is ghosting at the door of Starbucks. This is due to insufficient light so that features can not be detected well.

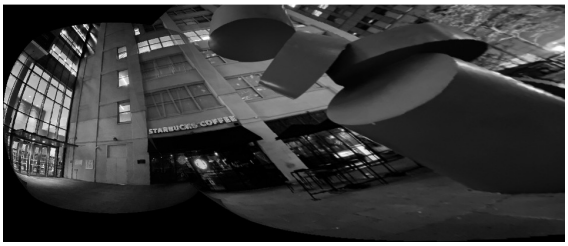

Fig. 5. Stitching Result of MetroTech

5) Stitching Results of Roosevelt Island: Taught lesson from the last two experiments, we decided to take pictures at an open place in daytime. So the depth of scenes are relatively close and light is strong enough to detect features. The result is nearly excellent and is shown in Fig. 6. Although the final stitched image is not a perfect rectangle, it is reasonable because the photos were taken by hand and slight shake could not be avoided. This issue is able to be addressed by taking advantage of a tripod. However, we do not have one so this result is the best result we can achieve using our current device.

Fig. 6. Stitching Result of Roosevelt Island

B. Spherical Panorama

1) Test Images: The experiments of spherical panorama uses a set of pictures taken from iPhone 12 or iPhone 12 Pro (both focal length = 26mm). The information about the tests is in the table below.

Location	Depth	Time
Tandon Campus	Far	Afternoon
Inside Library	Close	Indoor
Library	Far	Afternoon

2) Stitching Results of Tandon Campus: As shown in Fig. 7, three pictures are stitched together. The shapes of the buildings are not twisted dramatically and the stitched points are not very obvious. However, some ghosting appears at one of the buildings. This is because shifts while taking photos.

Fig. 7. Stitching Result of Tandon Campus

3) Stitching Results of Inside Library: This test aims to stitch four pictures to form a 180 degree picture. Firstly, two pictures on the left will be stitched Fig. 8. These two images remain in normal shapes.

Then the the next two pictures on the right will be stitched together Fig. 9. The second picture is compressed too much, due to insufficient overlapping.

Fig. 8. Stitching Result of Desks of Library

Fig. 9. Stitching Result of Desks of Library

At last, we will stitch the two parts of the picture to form the panoramic image in Fig. 10. The problem of the result is that the left part is blended too much and the right part is too small due to over compression.

Fig. 10. Stitching Result of Desks of Library

4) Stitching Results of Dibner Library: In this experiment, we used eight pictures as the input. In order to make the most middle picture as the center of the result, we applied flip operation. In Fig. 10, the middle and right parts of the result remain in original shape, but the left part of the result was twisted. This may be resulted from the the directions of camera of picture 3 or 4 has changed dramatically. Another reason is the tested pictures are taken by changing the horizontal directions. Vertical directions remained the same while taking photos. In order to have a better result from spherical projection, we should have changed vertical directions when took photos. Due to time limitation, we didn't take pictures cover 360 degrees, this result is the best we can got by using spherical stitching.

Fig. 11. Stitching Result of Dibner Library

V. Summary

From this project, we reviewed the technology on traditional image stitching and learnt knowledge on panorama stitching. We fulfilled cylindrical panorama and did part job of spherical panorama due to time limitation. In the future work, we will finish the remaining part of spherical panorama.

References

[1] R. Szeliski *et al.*, "Image alignment and stitching: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2007.

[2] L. Wei, Z. Zhong, C. Lang, and Z. Yi, "A survey on image and video stitching," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 1, pp. 55–83, 2019.

[3] V. Kaynig, B. Fischer, and J. M. Buhmann, "Probabilistic image registration and anomaly detection by nonlinear warping," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[4] C. Harris, M. Stephens *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[7] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[8] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on pattern analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1150–1163, 2006.

[9] zwx1995zwx, *Principal of Cylindrical Projection*, 2018, available at https://blog.csdn.net/zwx1995zwx/article/details/81005454.

[10] FarryNiu, *Use opencv-python to make spherical projection of pictures*, 2022, available at https://blog.csdn.net/qq_43474959/article/details/108394740.