1 Changes from proposal

Original features proposed in proposal:

- Input of the program should be multiple photographic images and output should be panorama of these images.
- The program could handle images in random orientations.
- The program could handle images in random orders.
- The number of input images the program could handle is greater or equal than 2 and have no upper bound.
- The program could process the shape of final output panorama in regular shapes like rectangles instead of irregular shapes which looks unpleasing.

The first point is achieved. The program input is multiple images and the output is panorama of these images. The second point is applied for input images with original shapes for stitching, not for cylindrical panorama which may have bad stitching results. The third point which is random input order is changed to the order of images which is taken from left to right. The forth point is not changed and the fifth point is changed to the program could use cylindrical panorama to make images look natural. Because processing the output panorama into regular shapes may make the final output loss lots of information compared to previous image without shape fixing.

2 Key features and implementation

2.1 Cylindrical panorama stitching

To make the final stitched image from input images comfortable to see, it is essential to consider how to stitch these images. Figure 1 shows four input images for stitching. If image stitching is done using original rectangular input images, after transformation of these input images, warping the images and blending, the final stitched image may look like Figure 2, where the right part of the panorama looks strange and large, which is like being stretched. Moreover, the right panorama part occupies more resolution than the left part which makes the left panorama part look small.

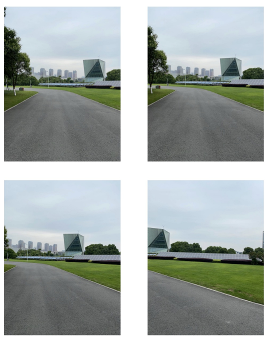

Figure 1: Input images for stitching

Cylindrical panorama could make the stitched image has even resolution in each stitched part which looks comfortable and without extreme part as shown in Figure 3 [1]. To do cylindrical panorama for input images, first input images should be transformed using cylindrical projection. Figure 4 shows input images in Figure 1 are transformed using cylindrical projection for stitching.

To convert the original image coordinates into cylindrical projection image, first the distance between the image and the center of the cylinder should be known which is focal length f [2]. Here the f is the distance between the cylinder center and the image center and these two

[1] https://www.cambridgeincolour.com/tutorials/image-projections.htm
[2] https://www.csie.ntu.edu.tw/ cyy/courses/vfx/10spring/lectures/handouts/lec07_stitching_4up.pdf

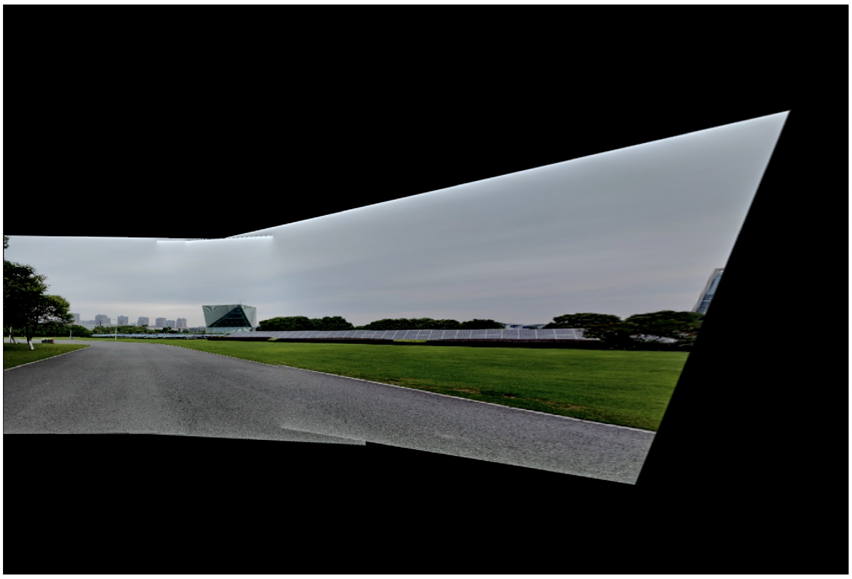

Figure 2: Image stitching without cylindrical panorama (original rectangular shape)

Figure 3: Image stitching with cylindrical panorama

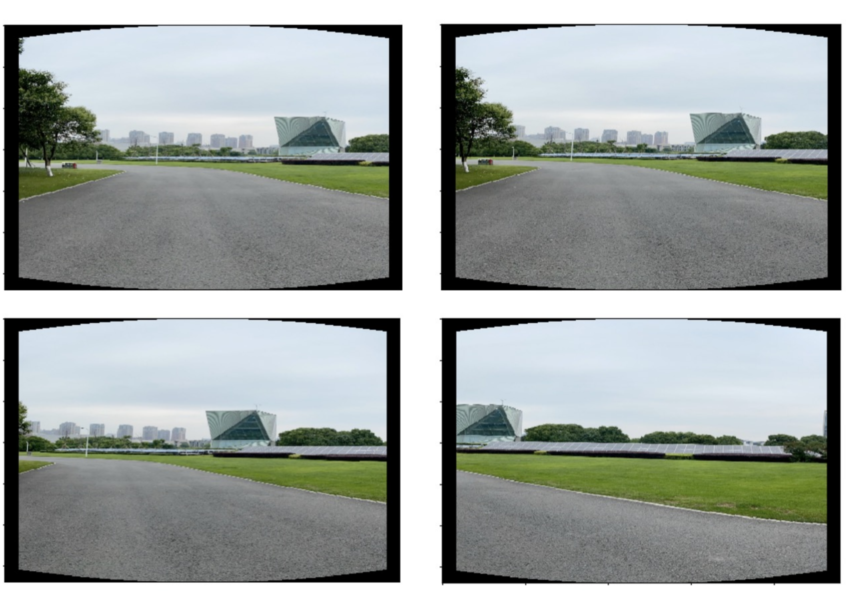

Figure 4: Input images cylindrical warping for cylindrical panorama

points are on the same horizontal plane. Because the camera parameters are unknown, the focal length is set to 500. Figure 5 shows the relations between the image and the cylinder. For each pixel position in the image, the angle θ between the pixel and f which horizontally which is $\arctan(x/f)$ where x is the distance between the x coordinate of the pixel and x coordinate of the center of the image. Using small angle approximation[3], the angle could be roughly approximated as x/f.

The position of image pixel in cylinder of radius 1 using cylindrical coordinates is then could

[3]https://en.wikipedia.org/wiki/Small-angle_approximation

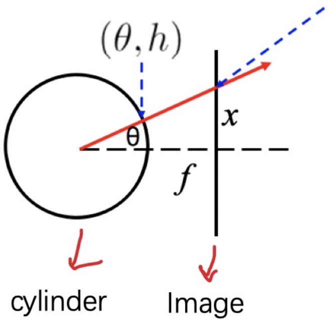

Figure 5: relationship for cylinder and the image in cylindrical warping

be represented using (θ,h) where θ is the angle between f and the line projected horizontally using the line between the pixel and the cylinder center, and h is $y/\sqrt{x^2+f^2}$. The projected point in the x-y-z coordinates for cylinder of radius 1 shown in Figure 6 is then calculated as ($sin(\theta)$,h,$cos(\theta)$). The projection effects are better when the focal length equals to the radius of the cylinder.

It is better and more accurate to do inverse warping to warp the input image using cylindrical projection. For each point in cylindrical x-y-z coordinate of focal length and cylinder radius f, the corresponding x position is $fsin(\theta)$ and the corresponding y position is fh.

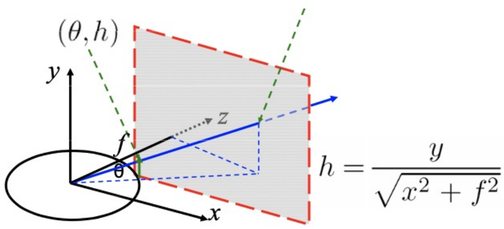

Figure 6: Transform original image pixels to cylinder

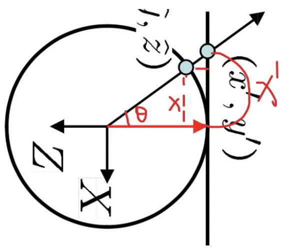

Figure 7: calculate positions for inverse warping from cylinder to original image

For inverse mapping, to get the corresponding pixel position on original input images, x and y coordinates on original input image should be got. To get x position on original image for the cylindrical warping image, because we know the angel between the focal line and the line between the focal center and the point on the cylinder is θ, we can get the corresponding x on cylinder x-y-z coordinates which is $fsin(\theta)$. Assume x' is the distance between the image center and the corresponding x coordinate of original image. Then $f/fcos(\theta) = x'/x$, which is $f/fcos(\theta) = x'/fsin(\theta)$. After simplification we can get $x' = fsin(\theta)/cos(\theta)$. Because the x center of the original image is w/2 where w is the width of the image, the final corresponding x position of original image is $fsin(\theta)/cos(\theta) + w/2$. And the calculation of corresponding y position of original image is the same as calculating x position. In my program, for each point, the following matrix multiplication is applied. Because the final point position is 2D, we should divide the outcome by $cos(\theta)$ which is $x[2]$.

$$x = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin(\theta) \\ h \\ \cos(\theta) \end{bmatrix}$$

$$x = x/x[2]$$

Figure 8: calculation matrix for inverse warping in my program

2.2 Pyramid blending

Pyramid blending is applied to make the stitched image look more natural in overlapping area. It is done for every two images which have overlapping. First, Gaussian pyramid is built for these two images. For each image, first gaussian filter is applied, then downsampling is used for the filtered image to reduce the size to 1/2 of the original one. For gaussian pyramid of multiple layers, the upper layer G_l is got by do gaussian filtering of the lower layer G_{l-1} then do downsampling like $G_l = \text{REDUCE}(G_{l-1})$. For downsampling operation, I used 'pyrDown(img)' in opencv.

Then two laplacian pyramids is built for the two gaussian pyramids of the two images. For each gaussian pyramid, the uppermost level is regarded as the uppermost level of laplacian pyramid. Each lower level of the pyramid is built using the subtraction of the level and its upper level. For each level except the uppermost one, the size is 2 times of its upper level. So the subtraction is done using the map of the current level and upsampling of the upper level to the size which is the same as the current level. Upsampling operation is make the upper level map size two times larger than its original size. For upsampling operation, I used 'pyrUp(img)' operation.

For width or height of an image of odd number, 'pyrDown(img)' to build gaussian pyramid will do rounding to integer. For example, if the width is 101, after downsampling, the upper level has width 101/2 which is 50.5. But using 'pyrDown(img)', the size is round to 51. When do laplacian pyramid, for level with width 101 in gaussian pyramid, we do upsampling for its upper level to do subtraction. But upsampled image of previous width 51 has width 102 now, which is not the same as image map in the level of gaussian pyramid with width 51. To prevent this, in my program, after upsampling of upper level image, there is a comparison between the map in gaussian pyramid and the upsampled map. If for updampled map, the width has one more pixel than the map in guassian pyramid, then the right most column is deleted to make the width the same as the map in gaussian pyramid of the same level. If the height of the upsampled map has one more pixel than the map in guassian pyramid, then the most bottom row is deleted to make the height the same as the map in gaussian pyramid of the same level.

For blending operation, the two laplacian pyramids for the two images are combined in each level according to mask. In this program, the input images are usually in the order taken from left to right. As Figure 9 shows, there is an overlapping region for two images(assume). Mask of each image is made. if we want some parts of an image to be in the final blended image, then in mask, these pixels in these parts are marked as 1 and 0 otherwise. The first for me to come out is for overlapping regions, one mask is all 0 for one image and the other mask is all 1 for another image. However, the blending effect is not good. For image which has mask of blending region as all 1, the final blended image has edge effect like Figure 10 shows.

From my point of view, it is better to design the method that for each mask, it shares some part of the overlapping region and the blending is from the middle part of the overlapping region. Figure 11 shows the idea that each mask shares some parts of overlapping region. For the red part's pixels, 1s are set in the mask of the left image and for the right orange part pixels, 1s are set in the mask of the right image. Figure 12 shows the blending results using the idea which is better and look natural compared to Figure 10. For each mask, the corresponding

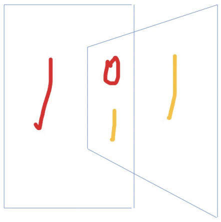

Figure 9: original masks I set for two images

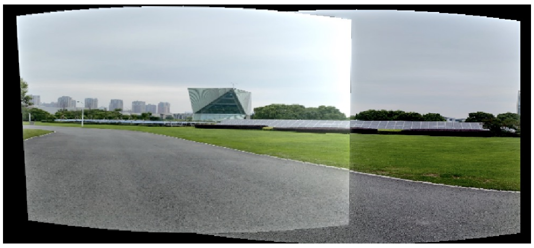

Figure 10: the stitched image using the mask set before

gaussian pyramid is built. For blending, each level is combined using corresponding level of masks of the two images and corresponding level of images in the two laplacian pyramids. For blending using pyramid blending, the more levels the pyramid has, the better blending effect the final stitched image has. So I set the level to 10.

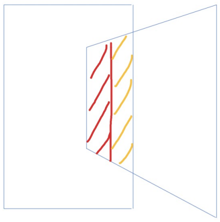

Figure 11: Newly designed masks to fix the problem in Figure 10

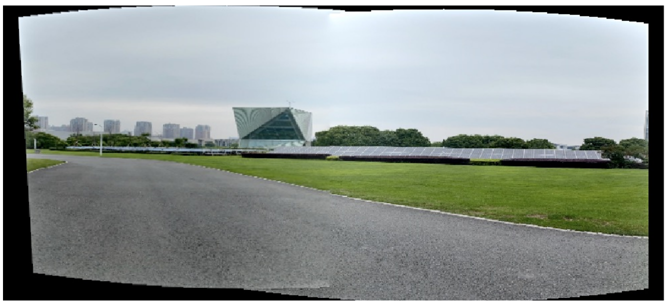

Figure 12: the stitched image using the newly designed masks

2.3 Find matches between two images and warping for stitching

To find the matching points of two images, SIFT is used. For SIFT, there are four steps. First find points whose surrounding patches are distinctive and for the points of an image, translmation invariance is calculated by calculated 'summing up the squared differences (SSD)' and using Taylor series expansion to rewrite the SSD formula and solve the hessian matrix formed in the rewrite SSD formula by finding if the smallest eigenvalue is big enough. If it is, the point could be chosen otherwise throw out. Then for each point, finding its main direction and rotate the patch in that direction by the the eigenvector which has the larger eigen value. Finally, SIFT descriptor is created no matter illumination conditions by creating histogram of edges orientations as the feature vector of the point. SIFT is applied using the function in opencv 'cv2.xfeatures2d.SIFT_create()'.

The homography is found using RANSAC. First, four feature point pairs are selected randomly to compute a homography. Then for all feature point pairs, to calculate if the transformed point pairs are close in a threshold and to see if the number if feature point pairs are high. Select the homography with the most of feature point pairs and calculate the transformation of input image. The homography which is used to warp one image to other shapes to stitch two images is found using 'cv2.findHomography'.

2.4 Stitching more than 2 images and input images with order from left to right

Originally I wanted to do stitching in random orders regardless left and right orders, but the final outcome is not good sometimes. Moreover, it is a little hard to do random order stitching and to judge whether the images are in right or left. So for better stitching effect, the input images are in the order of images taken from left to right and the stitching is based on the order of the images as Figure 13 shows. In Figure 13, 1, 2, 3 and 4 represent image 1, 2, 3 and 4 taken from left to right. The four images are divided into left images and right images. Image 1 and 2 are divided into left images and image 3 and 4 are divided into right images. These two kinds of images do stitching separately first then the stitched left images and the stitched right images are stitched together to form a panorama. In Figure 13, image 1 and image 2 are stitched together to form a new stitched image 5. Image 3 and image 4 are stitched together to form a new stitched image 6. Then the final panorama is got by stitching image 5 and image 6. Doing stitching for small set of images first could make the final panorama shape look smoother and have less extreme edges like Figure 2.

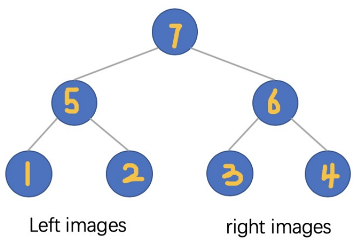

Figure 13: Stitching order for the input images

2.5 Choose whether to use cylindrical warping, whether to use resize operation, whether to set resize values

In my program, to compare the effect of image warping using cylindrical warping and stitching using original image, I set the choice whether to use cylindrical warping. Figure 14 shows the command line that run the program and whether to use cylindrical warping or not. The cylindrical warping setting is underline in red. '1' is to use cylindrical warping and '0' is not. If the input of the setting is not 1 or 0, the default setting is 1 which means to use cylindrical warping.

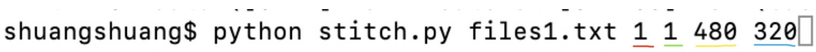

Figure 14: Input images processed for cylindrical panorama

For input images with larger resolution, the stitching effect may not be good and the time for computation is long. To cope with the problem, the choice whether to use resizing operation of input images is set which is underlined in green in Figure 14. If the choice is 1, then the input images are resized. The resized size could be decided in the yellow line and the blue line. The number in yellow line indicates the height of the resized image and the number in blue line indicates the width of the resized image. In the example in Figure 14, the expected resize size is height 480 and width 320. If the resized size is not set and the choice is resizing, the default is height 500 and width 400.

3 Strengths and weaknesses from results obtained

3.1 Strengths

3.1.1 Stitched images look even distributed using cylindrical panorama

Cylindrical warping is applied in my program. In my program, there are two options of processing the input images: using cylindrical warping and original images to do a comparison of image stitching using these two methods. Figure 15 shows four input images and the input image order is from top to bottom and from left to right. The left image of Figure 16 shows stitching of the four images without cylindrical warping of the four input images and it is clear to see that the rightmost part of the image is like being stretched and looks unnatural. Moreover, The parts of the stitched image in Figure 16 do not even distribute. The right image of Figure 16 shows the stitched image of the four inputs using cylindrical warping. From Figure 17, the parts of the stitched image are evenly distributed and the rightmost part looks natural. The final output panorama in my program should looks natural and has few distortions in the shape of the image, so cylindrical warping is applied. Figure 18 shows the cylindrical panorama using 7 input images and Figure 19 shows the cylindrical panorama using 9 input images.

Figure 15: Input images

Figure 16: comparison between the stitched images not using cylindrical warping and using cylindrical warping

3.1.2 Pyramid blending and blending mask setting

Pyramid blending was applied in my program as section 2.2 states. And the masks designed for the input images with overlapping area are also important. If the masks of images are designed as Figure 9 shows, where the mask values of overlapping region are all set to 1 in a certain mask and another mask set to 0 of overlapping region, the stitched image will have obvious gap effect as the left image in Figure 17 shows. Then the masks are designed as follows: for the left half part of overlapping region, the mask values of the left input image in this part are set to 1 and the right half part values are set to 0. For the right half part of the overlapping region, the mask values of the right input image in this part are set to 1 and the left half part of the overlapping region set to 0. After applying the two masks, the final outcome is like the right image in Figure 17, which looks better and more like from one image compared to the left image in Figure 17. Figure 18 shows image stitching using 7 input images and Figure 19 shows image stitching using 9 input images. And these two stitched images look natural.

Figure 17: comparison of panoramas using different masks

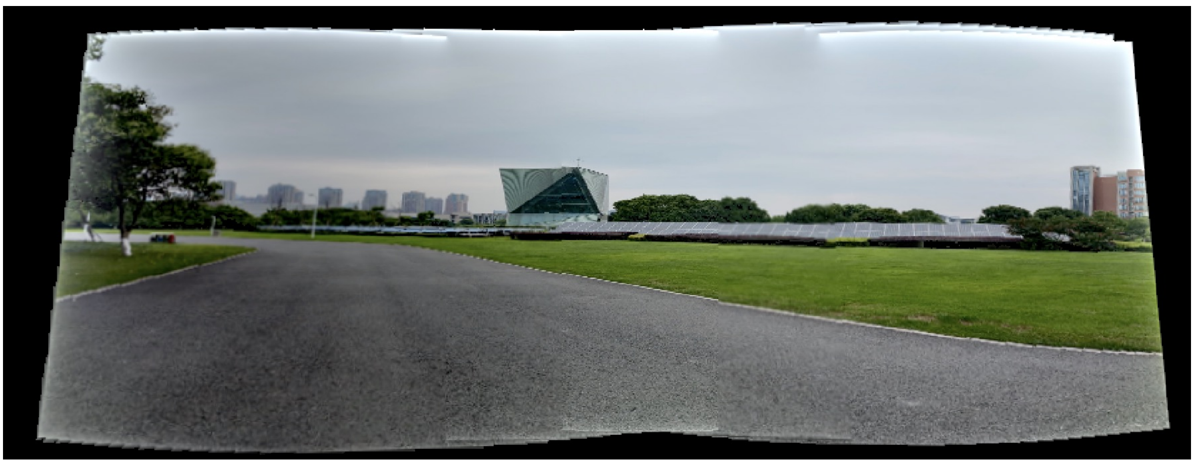

Figure 18: Cylindrical panorama using 7 input images(input images are resized to 480 * 320)

3.1.3 Resizing operations for images

For stitching of large-size images, the time to process the images and stitching takes lots of computation time. Thus, resizing option is used in my program to decide whether to reduce images into smaller sizes for example 480×320 for faster stitching. Figure 17 shows the stitched images where input images are resized to 480×320 and Figure 17 shows the stitched images where input images are resized to 500×400 before stitching.

Figure 19: Cylindrical panorama using 9 input images(input images are resized to 500 * 400)

3.2 Weaknesses

3.2.1 Moving objects could not be guaranteed well in stitched image

The effect of image stitching could be good when taking images of static objects like buildings, grass and trees. If images contain people who is walking, the final stitched image may look strange if the person is in the center of overlapping area. In that case, the masks may be made like in Figure 11 that only part of the people and the final stitched image may look unnatural because only a few parts of people are in the final image. In Figure 19, in the middle part of the square, it could be seen if carefully that then are some body part of another person with arm and leg which looks weird.

3.2.2 Blending part is a little blurring

Pyramid blending is applied for two input images to stitch together. There are two masks for the two images to decide which parts to combine together. However, in the middle which is the blend region, the stitched image after blending has more blurring effects than other parts. In Figure 19, on the middle right side of the image, there is some blurring effect on blending area if observe carefully. Although for laplacian pyramid, more levels could get better blending effects, the blurring effects on the blend place could also be noticed.

3.2.3 Handling input images which have big difference in taken angle and position

For input images, the orientations and the positions of two adjacent two images are taken should not be different too much. If two images are taken with very different orientations and positions, then the stitched image could be out of shape which may look not very comfortable and unnatural. Figure 20 shows three input images for stitching. And these three images were taken in the order from left to right with variations of angle not too much. The leftmost and the rightmost images have big variations regarding image taken angle and illumination.

The left image in Figure 21 shows the stitched image using the leftmost and the rightmost images in Figure 20. The right image in Figure 21 shows the stitched image using all the three images shown in Figure 20. The shape of the left stitched image in Figure 21 is a little distorted in the middle part because the image taken angle differs a lot and the illumination of the stitched image also looks unnatural in which the left part is much darker than the right part. From the right image of Figure 21, using three input images for adjacent two images the taken angle does not differ too much, the stitched image looks normal in shape of the buildings and the illumination condition varies gradually. Moreover, the shape of the right stitched image looks more comfortable than the left one.

Figure 20: 3 images in the order from left to right

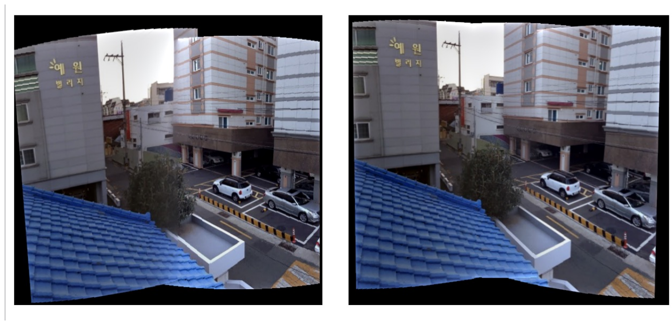

Figure 21: stitched image using the leftmost and the rightmost one in Figure 20 versus the stitched image using all the three images

3.2.4 Pixel information loss doing warping

For image stitching, there are some images to be warped to map another image. In this process, there is a transformation of the image to other shape. The warping is inverse warping and the pixel values are decided using interpolation which is not the same as previous original image. So the final stitched image may look with more blurring effects and the colors may be a little different.