Student Name: Chuyi *Hou* Student ID#: *1004197834*

Project Title:
Compare the performances of cylindrical warping and perspective warping when stitching a series of omnidirectional camera images taken by Clearpath Husky rover, while trying to reduce vignetting effects.

Project Problem Definition:

- **Problem**
  Given a series of omnidirectional camera images (5 images) taken by the rover [1], how to produce a rectangular-shaped 360˚ panorama, and how to reduce the vignetting effect? Moreover, compare the differences between results obtained using cylindrical warping and perspective warping. Is one warping performing better than the other given the same input data?

- **Changes from proposal**
  The problem had evolved from the one specified in the proposal. The cause of the change is due to algorithm implementation difficulties and time constraints. It is difficult to implement the Automatic Panorama Stitching algorithm [2] mentioned in the proposal since it requires several sophisticated algorithms, such as gain compensation, multi-band blending, and bundle adjustment for global optimization. I could not complete the implementation if I were to follow Lowe's paper given the project length. Therefore, I moved my focus to explore two warping methods, cylindrical warping, and perspective warping.

- **Relevance and Interests**
  This new problem is relevant because given that I am new to computer vision, I want to find some ways that are not too sophisticated and still can produce a decent 360˚ panorama. This way, I could get experience working with image processing and not be too stressful given current situation. Moreover, since the focus has been moved to the comparison between two warping methods, it is important to know when to choose one warping method over the other. Based on the configuration of the omni-directional cameras, along with the given camera intrinsic [1], I would expect that the cylindrical warping would perform better than perspective warping (Note: camera intrinsic must be known for the cylindrical warping to work properly). Personally, I would like to explore different methods that can achieve the same goal, meanwhile, finding the more efficient and effective ones.

Project Methodology:

- **Project Detail and Algorithm Implementation**

Input: 5 images [2,4,6,8,0]

I. **Feature Matching**
   a) Use SIFT detector to obtain key points and descriptors
   b) Use ratio test to filter the matches
   c) Use FLANN matcher to find the k-th nearest neighours.
II. **Stitching**
   a) Stitch [4,6,8] using cylindrical warping with affine transformation.
   b) Use Laplacian blending to render the stitched image. Name this output as 'x'.
   c) Stitch [2,x,0] without transforming x into cylindrical coordinates.
   d) Call feature matching every time when warping two images
III. **Cropping and Histogram Equalization**

Output: A panorama

Figure 1. Part 1: Produce the panorama using only cylindrical warping.

Input: 5 images [2,4,6,8,0]

I. **Feature Matching**
   a) Same as part 1.
II. **Stitching**
   a) Stitch [6,4,8,0,2] recursively, i.e. start with a = 6, for b in [4,8,0,2], find H between a and b using the matcher, stitch a to b, and set a to the stitched image.
   b) Call feature matching every time when warping two images
III. **Cropping**

Output: A panorama

Figure 2. Part 2: Produce the panorama using only perspective warping.

Note: [2,4,6,8,0] corresponds to the images taken by the cameras labelled with these numbers.

The project has two major portions. Figure 1 and figure 2 listed the algorithm for part 1 and part 2 respectively. Part 1 is producing the panorama using the cylindrical warping. Part 2 is producing the panorama using the perspective warping. The results obtained from part 1 and part 2 are compared.

- ***Methodology***

I. sift_matchers.py

This file creates a feature matching class using OpenCV's SIFT detector and FLANN matching. Affine transformation matrix and homography transformation matrix transformations are computed with built-in RANSAC using OpenCV's estimateAffine2D, and findHomography respectively. The matcher function will return either an affine transformation matrix or homography transformation matrix based on the input argument. Moreover, before each feature detection, the input images will be preprocessed using the adaptive histogram equalization. This will help to detect more features for low contrast images.

II. main.py

This file imports the sift_matchers and creates a stitch class with various internal functions that complete the project. Table 1 below listed the functions and their descriptions.

Table 1. Internal functions and functional descriptions

Functions	*Description*
__init__(self, input, output)	*Initializes input images by reading them from the input folder, creates camera intrinsic matrices, and setup the output directory.*
cylindricalWarp(self, img, K)	*Returns the cylindrical warp for a given image and its corresponding camera intrinsic matrix K.*
Stich_images(self, baseImage, warpedImage, warpedImageMask)	*Stitch two images using a mask. Copy warpedImage into baseImage only if values in mask is 255. This is only used when doing cylindrical warping. This will help get rid of the gaps between images when using warpAffine.*
Laplacian_blending(self, img1, img2)	*Renders the stitched image using Laplacian blending.*
hisEqulColor(self, img)	*Performs histogram equalization on the cropped final panorama before outputting the result.*
stitchThree(self, img_list, warp)	*Takes in a list of images and stitch them together using above functions and the matcher. It will behave differently based on the warp type. There are two types: 'cylindricalWarp', 'perspectiveWarp'.*
stitching(self):	*Calls stitchThree three times. The first two times are for part 1, and the last time is for part 2. See figure 1 and figure 2 for details on how the stitching is performed.*

These two python scripts described above will be able to produce the panorama. For improving the quality of the panorama, Lowe's paper suggested using gain compensation to reduce the vignetting effect and use multi-band blending to eliminate it [2]. However, these methods are too complicated to work with. Instead of implementing those sophisticated algorithms, the hisEqulColor and the Laplacian_blending (an Image Pyramid Algorithm) is utilized to even the contrast across the image [3] and to reduce the vignetting effect [4].

- ***Libraries Required***

  OpenCV, Numpy

- ***Data Used***

  Run3: Human-readable (base) data
  Camera frame frame000187_2018_09_04_17_44_08_405404
  Omni_image{0 to 9} || cameras_intrinsics.txt

Project Evaluation and Results:

- ***Evaluation metrics***
  Since a panorama is hard to evaluate using quantitative metrics (no ground truth, size of the output may vary, no good quantitative comparison methods, etc.), the metric I chose to evaluate the outcome is visualizing the panorama. By using this qualitative metric, one can make heuristic comments on the outcome and can also make adjustments accordingly if needed to improve the panorama. For example, one can try out different image blending algorithms or filters to improve the reduction of the vignetting effect and/or flatten the contrast across the panorama.

  In addition, since the project focus has been changed to the comparison between two warping methods, another qualitative metric will be used to evaluate the comparison. That is comparing the stretchiness of the panoramas obtained using different methods. The less stretched the stitched image is, the better the method performs given the rover data. In other words, if each input image occupied roughly the same proportion of the panorama, then the performance is adequate.

- ***Justification of the choice of the metrics***
  There is no quantitative metric for this project, and I think I need to justify this. During the development of this project, my algorithm does not provide the real-time stitching ability which is mentioned in the 'relevance of the project section' in the proposal. In other words, neither part 1 nor part 2 takes less than a second to produce the panorama (not even close to real time). I think execution time being a quantitative metric is not very useful for the comparison in this case, because more importantly, this project is about visualization. Both warping methods utilize feature matching and computing relative transformation matrices using RANSAC. Moreover, part 1 being slower is caused by the Stich_images function that was used for stitching the cylindrical warped images. This function has nothing to do with the performance of cylindrical warping. I chose to use it because it was one of many ways to get rid of the gaps when stitching cylindrical warped images. I am certain that there are lots of ways to stitch without using this function. Hence, I could improve the time complexity for stitching using the cylindrical warping in the future.

  Next, for the qualitative metric (visualize the outcome), this is the kind of project that is required to be evaluated based on qualitative metrics. In addition, for this project, it is difficult to tell the quality of the panorama from pixel values. For things like vignetting effect, contrast difference, and brightness difference, they can be observed by looking at the outcome. If they cannot be observed, then the outcome should be in a very good shape.

- ***Result***

Figure 3. Cropped histogram equalized panorama produced by using the cylindrical warping.

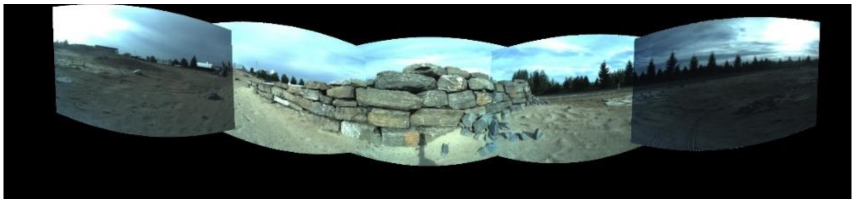

Figure 4. Uncropped and un-histogram-equalized panorama produced by using the cylindrical warping.

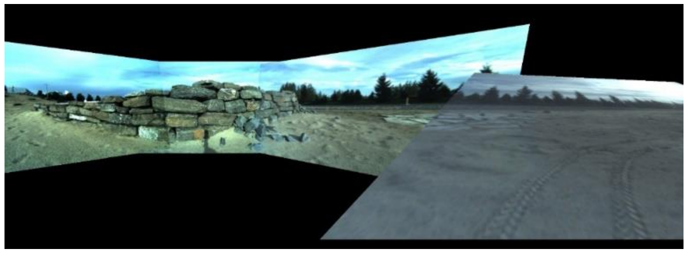

Figure 5. Not fully cropped panorama produced by using just perspective warping (un-histogram-equalized, no Laplacian blending and incomplete). Note: this image may or may not be produced when running the program, it is due to few matching points and the nature of the RANSAC algorithm.

- ***Discussion on the Result***
  The best result is presented in figure 3. I did not fully crop the image because that will reduce the view of the image too much. Figure 4 shows the uncropped and un-histogram-equalized version of figure 3. It is clearly shown that each input image occupies roughly the same proportion of the panorama. Moreover, it is noticeable that histogram equalization contributes to increasing the quality of the panorama. For example, the very left and right images become more sharpen and visible in figure 3 compare to figure 4, and the contrast for the panorama becomes more flattened.

  In addition, figure 5 demonstrates the result obtained by using just perspective warping. Notice that the panorama is incomplete. The reason for this is because the feature matcher was unable to find enough matching pairs to compute the homography. We can see that the images get more stretched out as they appear farther from the centre image. Since we are computing homography recursively, the error may accumulate. In other words, as we compute H further down in the loop, the current image that is going to be stitched will be even more stretched because the previous stitched image was stretched. In addition, the changes in image perspectives are large because of the camera configuration [1]. Hence, it makes the result less adequate. Whereas, in figure 4, we used cylindrical warping with affine transformations, and the sizes of the images being stitched stayed even.

  Therefore, we can conclude that the configuration of the cameras affects the choice of warping method to a large extent. For this project, it is better to use cylindrical warping to produce panoramas for the images collected by the rover. The reason behind stretching when using homography might be due to the perspective of the cameras being so diverged (only 5 cameras are covering a 360˚ view). Perspective warping should work better in the case where there are fewer changes in image perspectives (e.g. taking images with larger overlapping areas). The reason for cylindrical warping performing better, in this case, is because the omnidirectional cameras form a perfect cylinder plus the camera intrinsic is known.

  Moreover, the middle three images in figure 4 are merged into one image with few flaws, whereas obvious contrast differences and stitch marks are observed in the same three images in figure 5. Hence, we have gained insight into histogram equalization and Laplacian blending. They can help reduce the vignetting effect and improve the quality to some extent as desired.

- ***Challenges***
  For implementation:
  Initially, I found that feature matching for dark images was not working sufficiently well. There were little good matches. This issue was solved by preprocessing the image pair using adaptive histogram equalization before every call to the feature matching function.

  Then, I faced the challenge of stitching more than two images together. Initially, I thought I could find all the relative transformations between pairs of images beforehand. Then, when stitching, I could compute the transformations using the relative transformations, (i.e. $H_1^0 H_2^1 = H_2^0$). However, it did not work as I expected. Lots of stretching effects were introduced and error might *accumulate for further matrix multiplications. Afterward, I realized that I only need to do the feature matching and transformation calculation when stitching two images. This way, I could find the transformation between the stitched image and to-be-stitched images directly without performing matrix multiplication. This will reduce the chances of accumulation of errors in H.*

Next, global optimization was another challenge for producing a high-quality panorama with better views and better connectivity. Part 2 (perspective warping) result demonstrated that having all the homography transformations are not enough to produce a good panorama. Therefore, I found another warping, the cylindrical warping. It worked surprisingly well for the rover data. I think it is because that the way they are set up, they form a cylinder. Whereas perspective warping may require less changes in perspective so that the stretching effect can be reduced.

Moreover, to reduce the vignetting effect, one could use histogram equalization when rendering the panorama. Another way is to consider gain compensation combine with multiband blending [2]. Due to the implementation difficulty of the latter approach, I utilized the prior method, histogram equalization, for this project, and the result turns to be decent.

<u>For the evaluation:</u>
It was challenging to come up with a quantitative metric that could evaluate the outcome by outputting numerical values. There is no good ground truth to compare with. The stitched images that came with the dataset were not adequate because it was not the aim of this project (images were distorted and the vignetting effect was obvious). However, one could use the dataset stitched images to make heuristic comment on the performance of the project outcome.

- ***Conclusion***
  This project has been a very intellectually stimulating, yet challenging computer vision project for an introductory course. I had learned a lot by completing this project. Lots of literature reviewing took place during the proposing stage. Many refining and replanning happened during the development process. After completing this project, I strongly believe that my research and development skills are polished, and more background/basic knowledge for computer vision and insight on the state-of-arts algorithms are gained.

References:

[1] O. Lamarre, O. Limoyo, F. Marić, and J. Kelly, "The Canadian Planetary Emulation Terrain Energy-Aware Rover Navigation Dataset," The International Journal of Robotics Research, vol. 39, no. 6, pp. 641–650, 2020.

[2] Brown, M., Lowe, D.G. Automatic Panoramic Image Stitching using Invariant Features. Int J Comput Vision 74, 59–73 (2007). https://doi.org/10.1007/s11263-006-0002-3

[3] "Histograms - 2: Histogram Equalization," OpenCV. [Online]. Available: https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html. [Accessed: 10-Dec-2020].

[4] C. S. Verma and M. Ju, "Panoramic Image Mosaic," imagemosaic. [Online]. Available: http://pages.cs.wisc.edu/~csverma/CS766_09/ImageMosaic/imagemosaic.html. [Accessed: 08-Dec-2020].