For this project images, sample images were provided for three tasks. Task 1 involved stitching of two images t1_1 and t1_2, to form a single panoramic image. The important aspect of this task was to blend them in such a way that the foreground could be separated and masked. Task 2 involved generalizing the code written in Task1 in a way that a panorama consisting of multiple images could be created. Another aspect of this task involved using one-hot encoding to compare the input images and to determine which of them match. Lastly Task 3, includes utilizing images apart from the samples provided for creating a panorama.

Task 1

For this task, SIFT[1] was first used to extract the feature descriptors and keypoints for both input images. Based on these keypoints and descriptors, matches for pixel coordinate points in both images are identified. Two points are considered as a match if and only if their value is within a threshold.

Once the matches are identified, we then use Homography to identify the Homography/transformation matrix to morph the image. This morphed image is then stitched to get the panorama image. For this purpose, findHomography and warpPerpective\ methods from OpenCV[2] were used. An attempt to blend/blur the merged edge was made by using Laplacian blend/stemmingClone, etc.. However, due to incompatibility issues these were not implemented.

To remove the foreground from both images averaging of pixels was used to determine if the background is to be displayed or the foreground, Deghosting /Poisson Image Editing was considered for this purpose as well. However, due to issues with compatibility during implementation, these were not used.

Task 2

For this task, the logic from Task 1 was re-utilized. In order to create a panorama, In order to simplify the task, a seed image was identified. This image was then matched with the first image to create a warped intermediate output. Both the seed and the first image are then removed from the image array. The updated image is then stitched and merged with subsequent images. At each iteration, the updated image is modified. In order for this to succeed, it is assumed that each subsequent image will have enough keypoints to match the image in its current state. Apart from the transformations used in Task 1, perspective transformation has also been used.

Lastly, in order to compute the one-hot encoded array, each image in the array is matched with every other image including itself. An image is considered to match another image only if the number of matches is greater than a threshold. These results were then stored in the overlap output file.

Task 3

For this task, an attempt to utilize images clicked through a smartphone was used. These images have also been saved in the data folder. However, due to the huge image size and improper alignment, a decent stitch was not observed. An attempt to retake the images was made, however, due to bad weather (snowfall), this was not implemented. The assumption here was that the snow would act as noise which could further hamper the keypoint matching and panorama creation. Instead, sample images from Adobe's Panorama Dataset[3] were used for this task.

References

[1] SIFT (https://docs.opencv.org/master/da/df5/tutorial_py_sift_intro.html)

[2] OpenCV(https://docs.opencv.org/2.4/)
[3] Adobe Panorama Dataset (https://sourceforge.net/adobe/adobedatasets/panoramas/home/Home/)