John Radice
GTID 903095161
Assignment 8

1. The three images below were used to create the final panorama image. The attempt was to capture images to create a planar panorama:

Image Left

Image Middle

John Radice
GTID 903095161
Assignment 8

Image Right

2. Below is the final panorama image using overlapping average blending with 40 matches:

Final Panorama

John Radice
GTID 903095161
Assignment 8

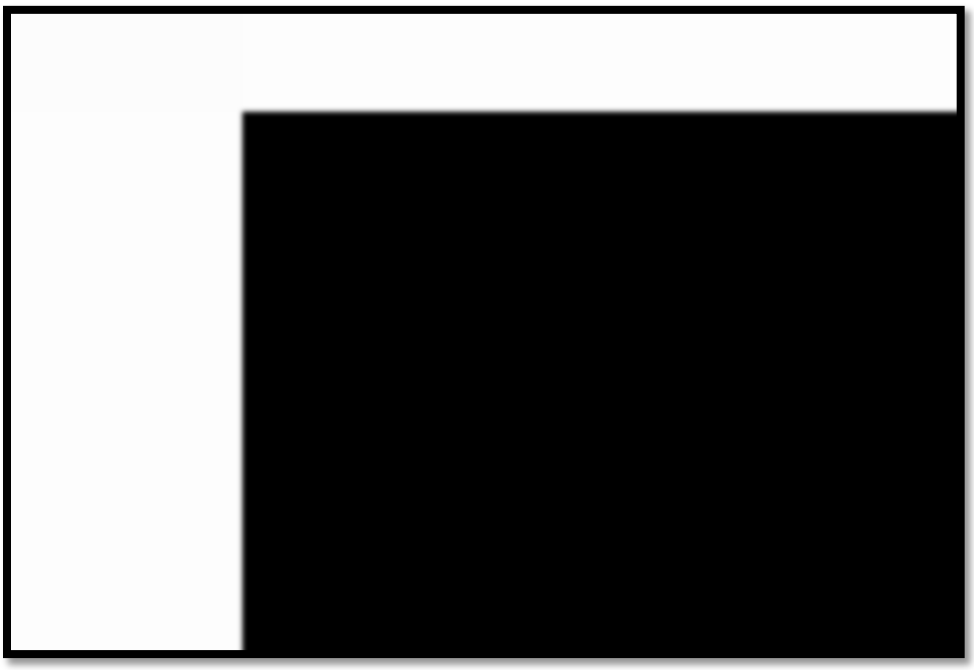
3. **blendedImagePair:**

Two attempts were made for the blendedImagePair implementation. The first was the basic averaging of the overlapped regions. The second an attempt to feather the overlapping edges. The feathering attempt involved creating a step function of the viewable region of the warped image. Once the step function was created it was then blurred using cv2.GaussianBlur with a 75 x 75 kernel. The blurred step function was then normalized and applied to the warped image. The inverted blurred step function was applied to image 2 as show below:

```
    Args:
        warped_image (numpy.ndarray): The image provided by cv2.warpPerspective.
        image_2 (numpy.ndarray): The image to insert into the warped image.
        point (numpy.ndarray): The point (x, y) to insert the image at.

    Returns:
        image: The warped image with image_2 blended into it.
    """
    output_image = np.copy(warped_image)
    image_2_copy = np.copy(image_2)
    image_2_expanded = np.zeros_like(warped_image)
    # REPLACE THIS WITH YOUR BLENDING CODE.
    # Feathering attempt
    image_2_expanded[point[1]:point[1] + image_2.shape[0],
                     point[0]:point[0] + image_2.shape[1]] = image_2
    # Create step functions
    step_x = np.zeros_like(warped_image)
    step_y = np.zeros_like(warped_image)
    step_x[: ,:point[0]] = 255
    step_y[:point[1], :] = 255

    step = step_x + step_y
    step[step > 255] = step[step > 255]*0.5

    # Blur step function
    blurred_step = cv2.GaussianBlur(step, (75,75), 0)

    blurred_step = blurred_step*0.00392157 # 1/255

    output_image = output_image*blurred_step

    image_2_expanded = image_2_expanded*(1 - blurred_step)

    output_image = output_image + image_2_expanded
```

Unfortunately the output of this portion caused an error during the second iteration in the findMatchesBetweenImages function. Specifically the sift.detectAndCompute function could not process the blended image. At this point no further attempt was made to process the octaves to complete the feathering algorithm. Below are images of the step function and the output of the first iteration:

John Radice
GTID 903095161
Assignment 8



Feathering Step Function

Image 1 and Image 2 “Feathered”

John Radice
GTID 903095161
Assignment 8

This resulted in falling back to the overlapping average implementation. For this implementation the overlapping regions were sliced and scaled using NumPy. Once the overlapping regions are calculated and scaled the scaled copy of image_2 is expanded and added to the scaled copy of the warped image as seen in the code below:

```
    output_image = np.copy warped_image)
    image_2_copy = np.copy image_2)
    image_2_expanded = np.zeros_like warped_image)
    # Average of overlap
    warped_overlap = output_image[point[1]:warped_image.shape[0],point[0]:warped_image.shape[1]]
    image_2_overlap = warped_overlap[:image_2.shape[0], :]
    boolean_overlap = image_2_overlap > 0
    output_image[point[1]:warped_image.shape[0],point[0]:warped_image.shape[1]] = \
        warped_image[point[1]:warped_image.shape[0], point[0]:warped_image.shape[1]]*0.5
    image_2_copy[boolean_overlap] = image_2_copy[boolean_overlap]*0.5
    image_2_expanded = np.zeros_like warped_image)
    image_2_expanded[point[1]:point[1] + image_2_copy.shape[0],
                     point[0]:point[0] + image_2_copy.shape[1]] = image_2_copy
    output_image = output_image + image_2_expanded

    return output_image
    # END OF FUNCTION
```

While this did create a panorama the blending seemed to have changed how the additional iterations compute the perspective warp compared to test code that pastes the second image into the panorama:

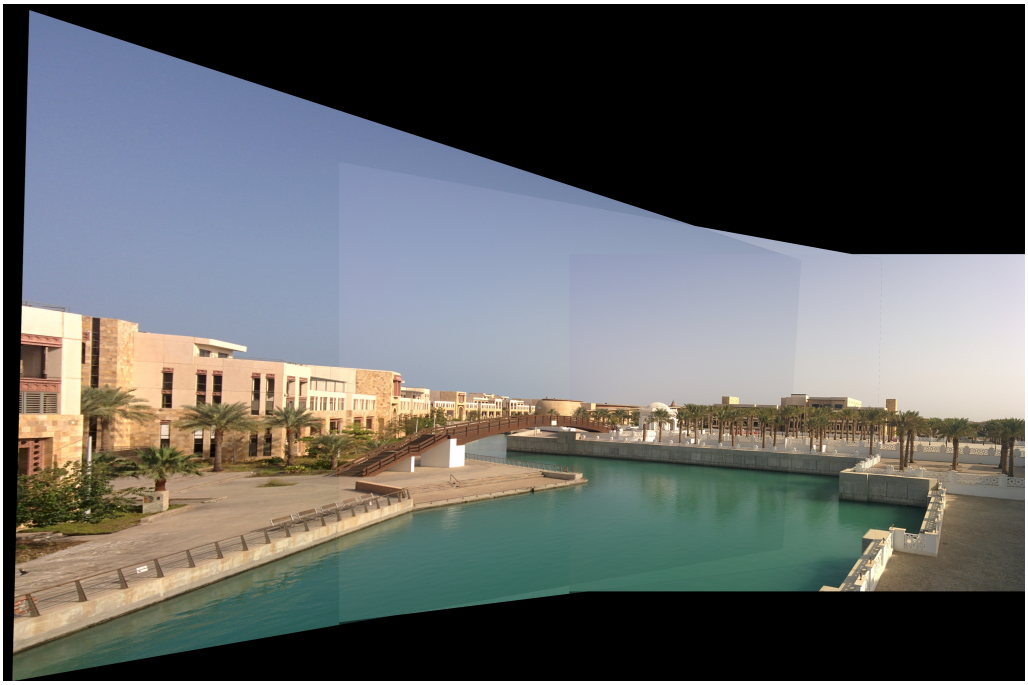
John Radice
GTID 903095161
Assignment 8

Paste Blending Panorama with 20 matches

Overlapping Average Panorama with 20 matches

The above image seems a bit too warped and has artifacts as well as signs for blurriness. However doubling the number of matches had a positive effect and helped remove some of the blurriness and artifacts:

John Radice
GTID 903095161
Assignment 8



Overlapping Average Panorama with 40 matches

In addition, reducing the matches results in no improvement. The blurring and artifacts are still present as well as the warping:

Overlapping Average Panorama with 5 matches

On my final panorama I used 40 matches to get a better quality image using the overlapping average blend.

John Radice
GTID 903095161
Assignment 8

I attempted to take a planar panorama as best I could. While my panorama seemed to blend better than the test images I wish I would've been able to implement feather down the seam to eliminate the visible edges between the separate photos.