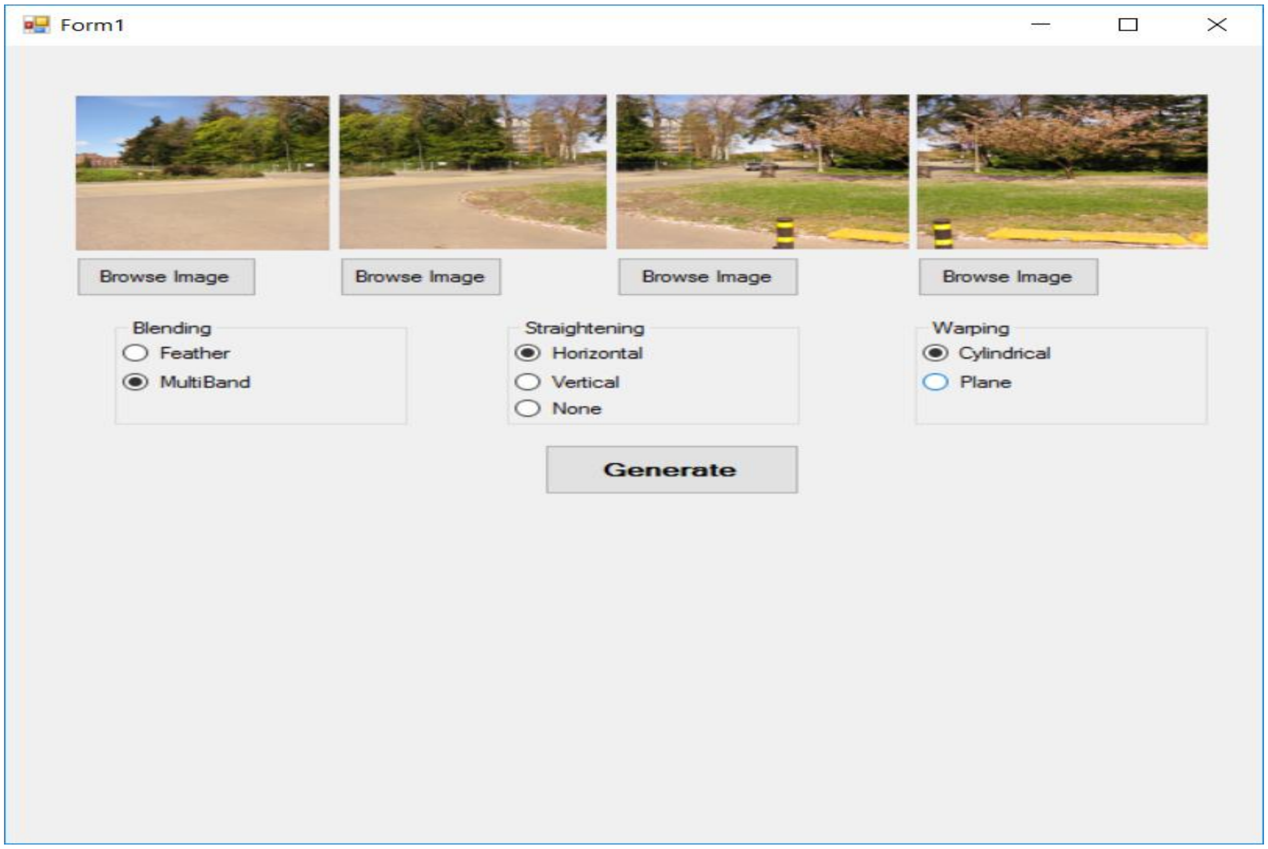
CS 510 Project #2: Panorama Synthesis

Submitted By: Hrishikesh Deshmukh

deshmukh@pdx.edu

The goal of this project is to implement a panoramic image synthesis system. I have used the libraries provided by OpenCV for implementation of this project. I have used Visual C++ as a programming language to implement the main algorithm for panorama synthesis and then I used C# to implement graphical user interface (GUI) to pass the different images to the panorama algorithm and select different blending, straightening and warping options. I am executing Visual C++ executable inside the C# code.

The application takes up to 4 images taken into sequence as an input. The images must be provided in the correct sequence to produce good quality results. User needs to browse up to 4 images from his computer like below:



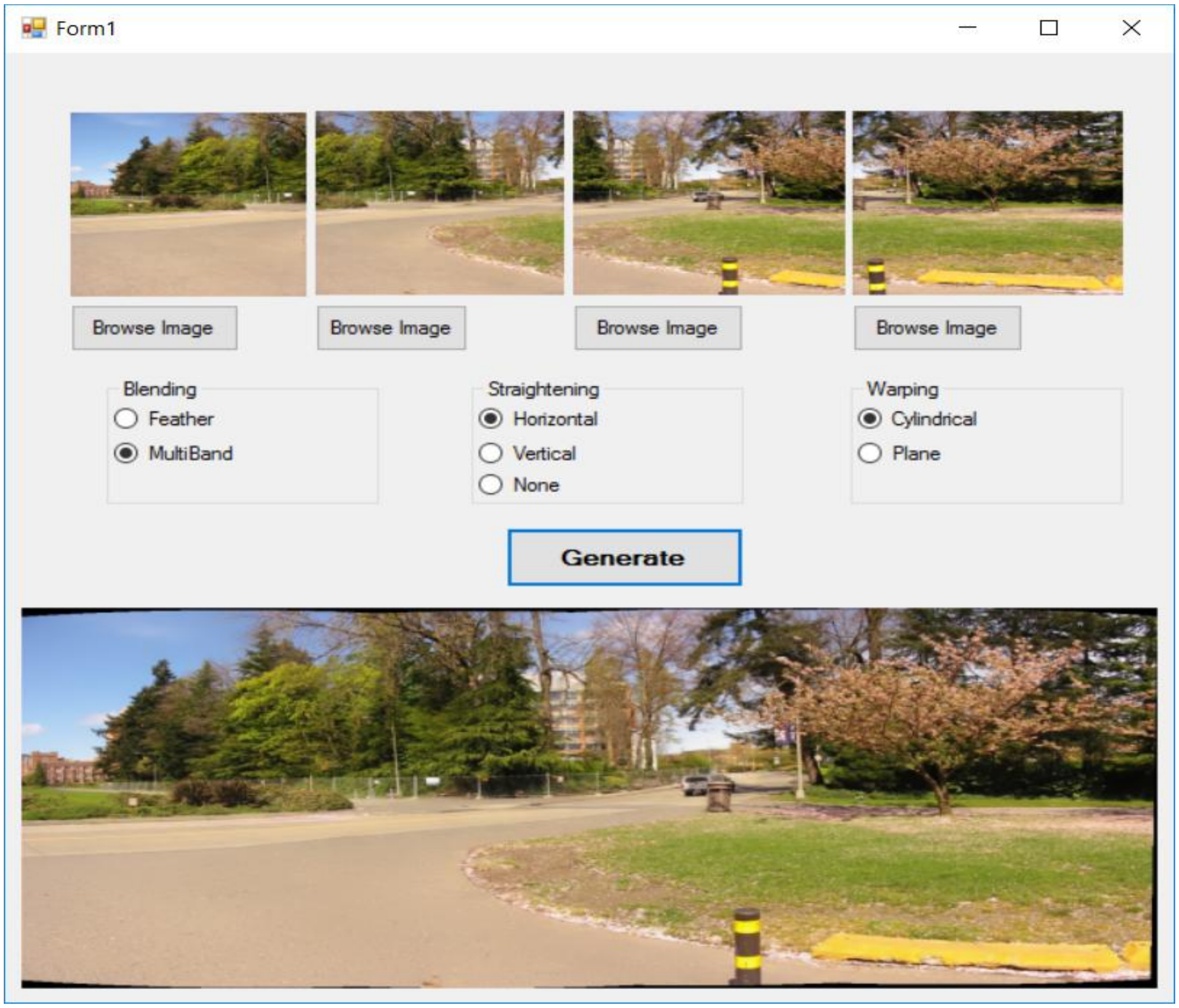
Blending: Feather [✓ MultiBand]

Straightening: [✓ Horizontal] Vertical None

Warping: [✓ Cylindrical] Plane

And then, user has to select from “Feather” and “Multiband” for Blending, “horizontal”, “Vertical” or “None” for straightening and “Cylindrical” and “Plane” for Warping options.

Once user uploads all the images and selects all necessary options, he has to click on the "Generate" button which generates the panorama image and is displayed in the picture box below the generate button like below:



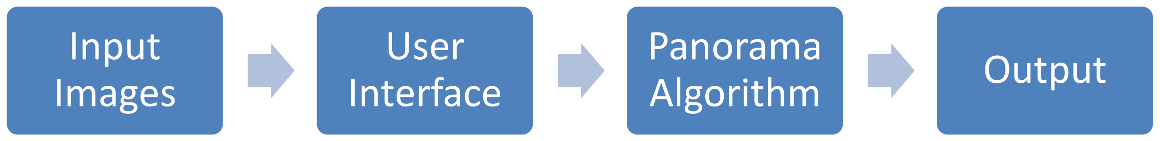
Blending: Feather [✓ MultiBand]

Straightening: [✓ Horizontal] Vertical None

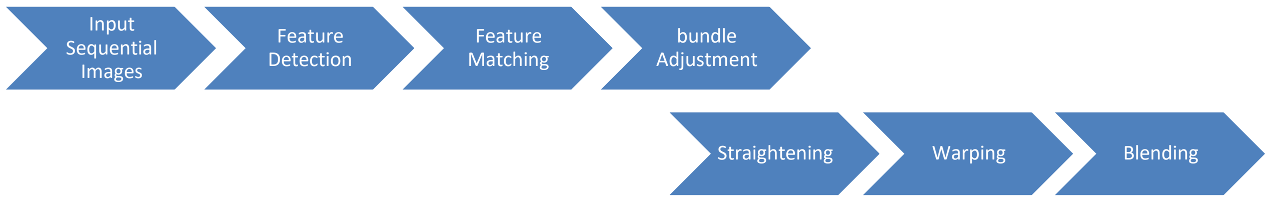
Warping: [✓ Cylindrical] Plane

As we can see, user has given four sequential images as input and has selected "Multiband" for blending, horizontal for straightening and cylindrical for Warping and the application has generated the panorama as per user's selection. The user interface passes images and user preferences to the main panorama synthesis algorithm, which stitches all the images, applies blending, straightening and warping options and produces the final resultant image and sends it back to the user interface. The user interface then shows the result on the screen.

The detailed flow of the application is explained on the next page.



Steps involved in Panorama Synthesis:



1. Input Images

The user selects images from user interface and passes it to the main algorithm. In the main algorithm, there is imread() function from OpenCV library which reads the input images and saves them as matrices.

2. Feature Detection

This is done by using SURF feature detector of OpenCV. The function SurfFeatureFinder() from OpenCV libraries detects all the features of input images and stores them in an vector called `ImageFeatures`.

3. Feature Matching

This is done by using `cv::detail::BestOf2NearestMatcher`, a matcher in OpenCV libraries which 2 nearest matches between features and selects only the best match between them.

4. Bundle Adjustment:

This step calculates the camera rotation and camera positions. If the images were clicked using different angles, this step calculates the rotational axis of the camera and provides the rotational ratio between images to next steps to be successful.

5. **Straightening:**

The input images might have taken from different angles, what this steps does is, takes the rotational ratio provided by bundle adjustment and tries to rotate the image so it straightens up and looks well oriented. This is done by using wave correction. It uses the relative rotational matrices and rotates the images to bring them at the linear level.

6. **Warping:**

This step warps the image in different warper classes provided by OpenCV such as cylindrical, plane, etc. It makes use of rotational matrices and ratios calculated in bundle adjustment and straightening.

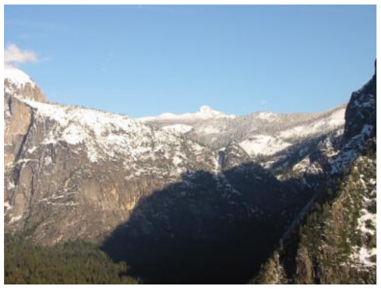
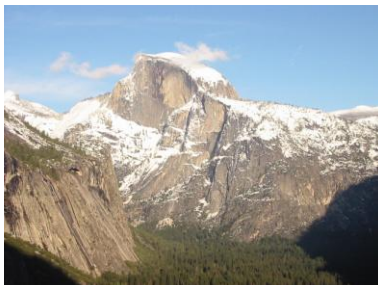
7. Blending:

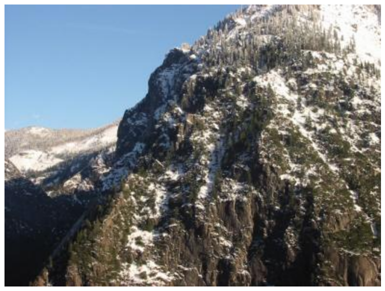
This step blends the stitched images together so that the color differences are not visible and the image looks uniform in nature. There are two types of blending types available in OpenCV i.e. Feather blend and Multiband blend. Multiband bend is more effective than feather blend as it blends the stitched images very nicely so that the difference between two images is not seen. In feather blend, the lines appear at the blending regions.

Example:

Blending Examples:

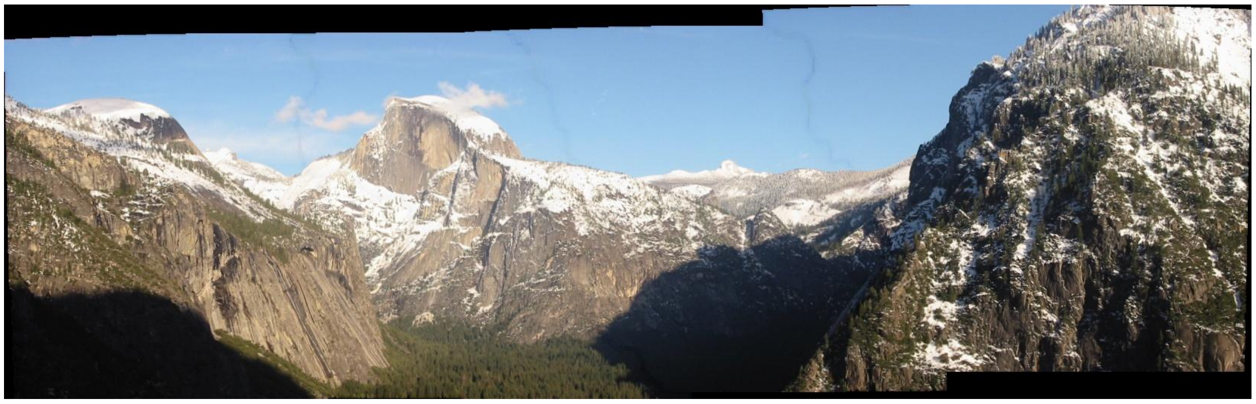
Input Images:



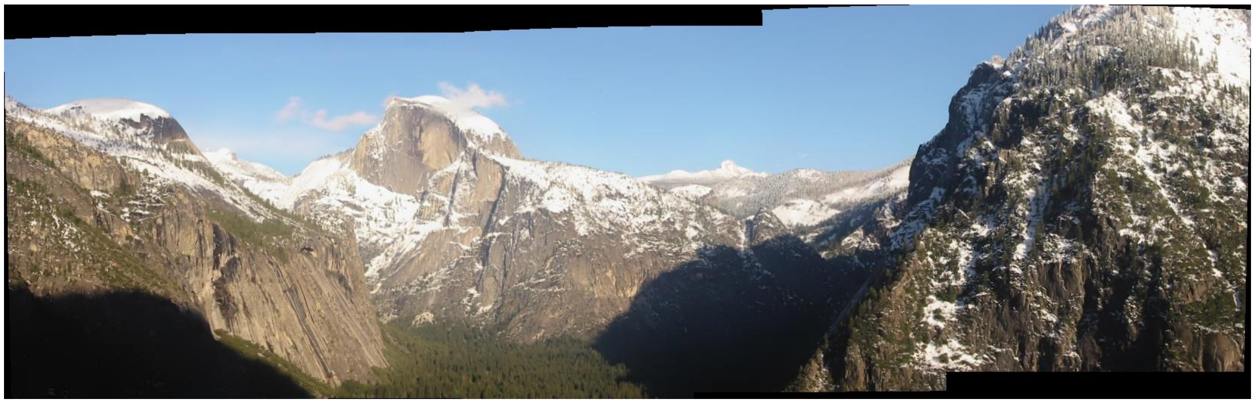
Output:

i) Feather Blending:



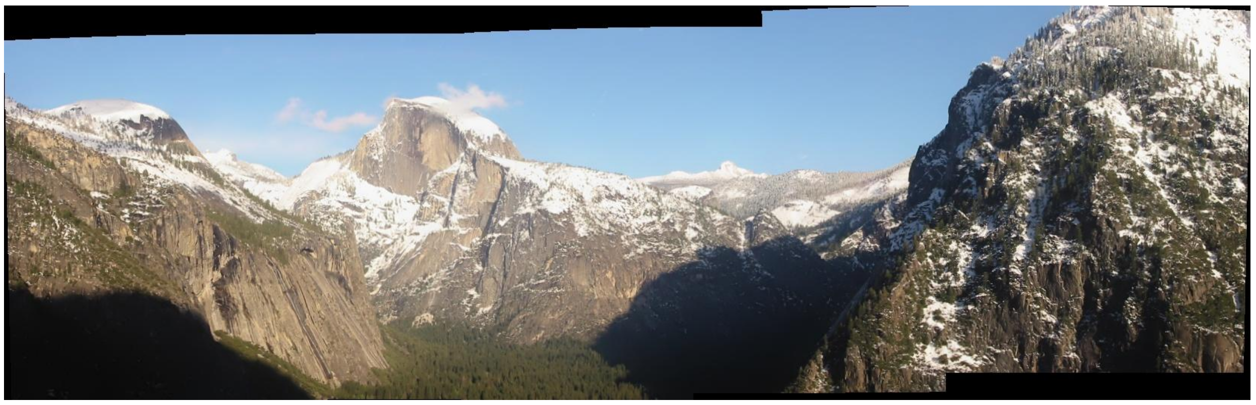
We can see the visible lines in the sky with feather blending.

ii) Multiband Bending

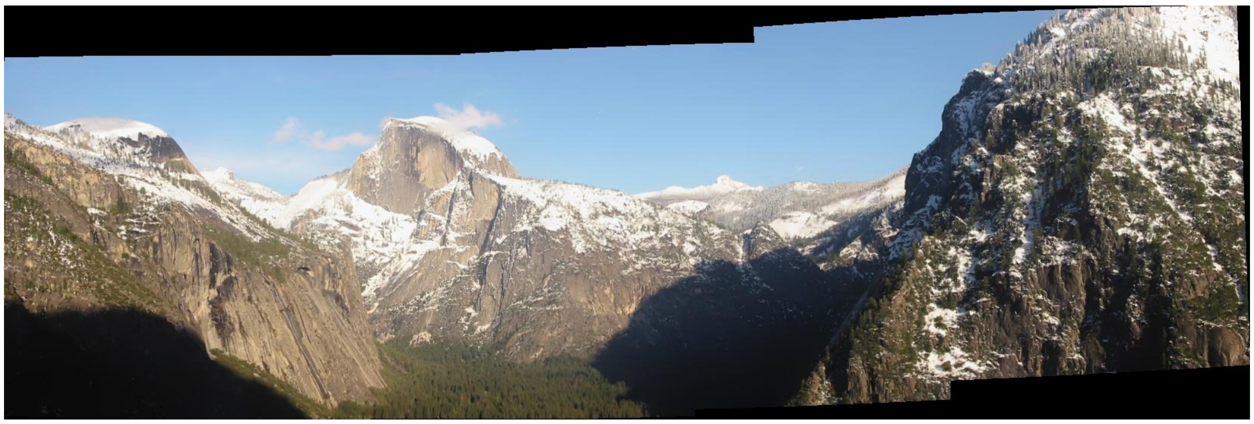


Straightening example:

i) With Straightening:



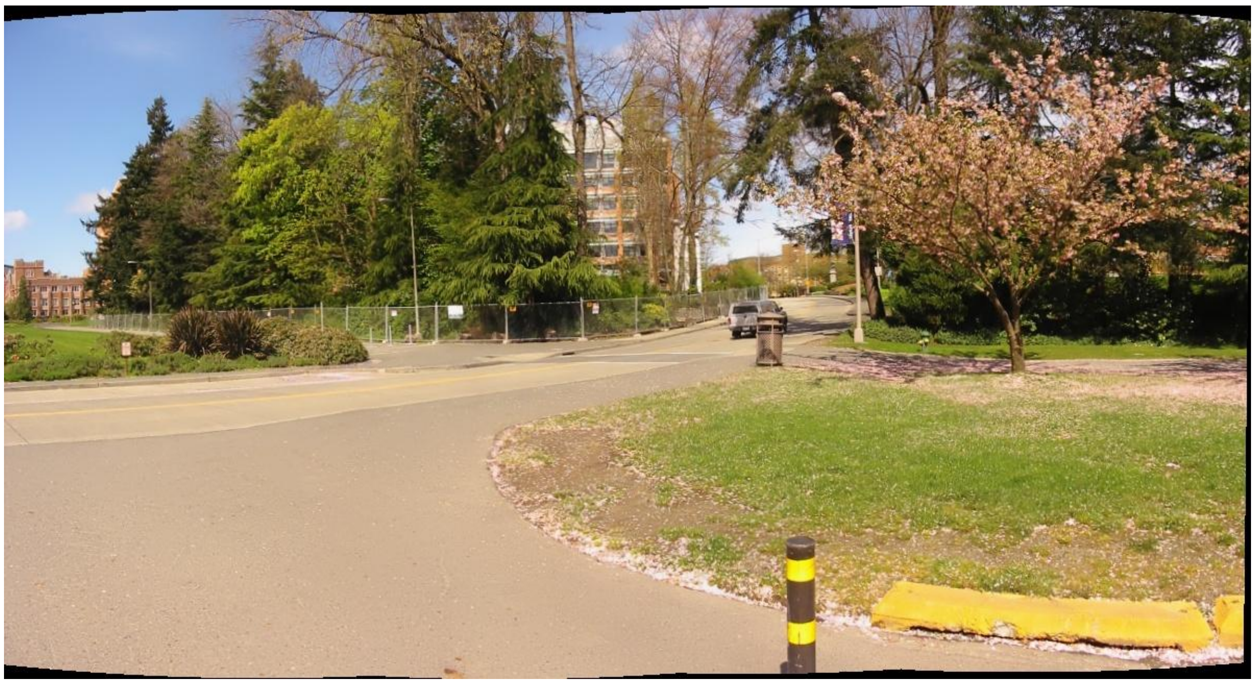
ii) Without straightening:



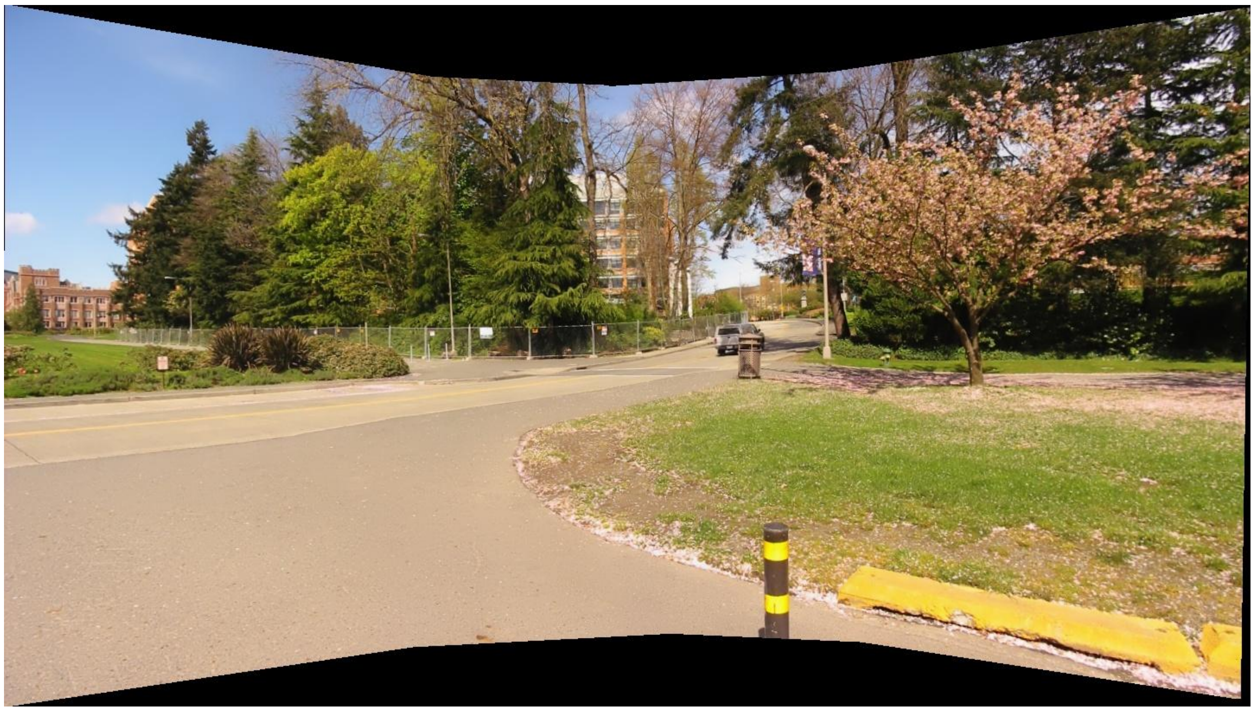
Warping Example:

Input Images:

i) Cylindrical Warping:

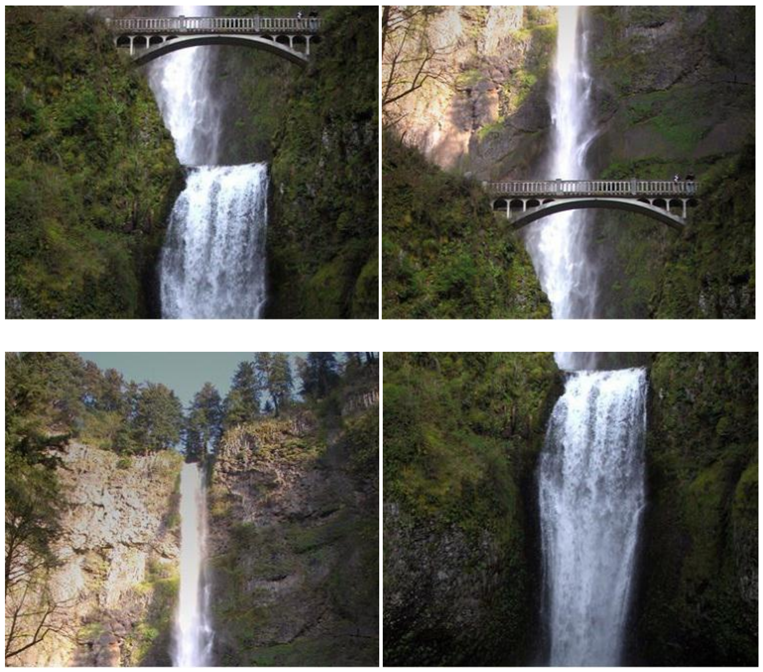


ii) Plane Warping:



Vertical Stitching and Straightening Example:

Input images:



Output: