CS 6475 Course Portfolio

Niraj Kumar

Fall 2020

Assignment 1: Camera Obscura

Goal: Build a camera obscura by letting light pass into the room through a tiny hole.

Purpose:

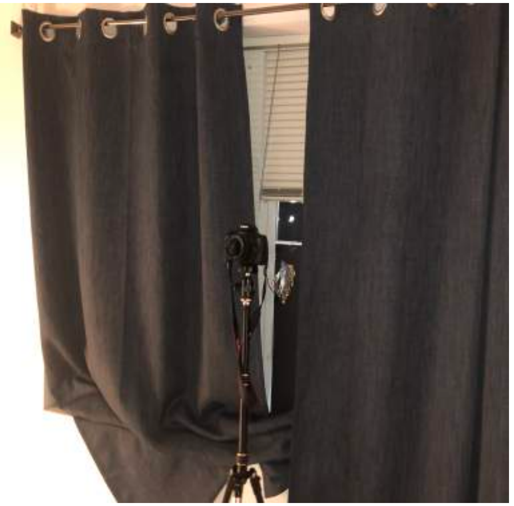
- ✓Understand how the pinhole camera serves as a basic model of all cameras.
- ✓building a room-sized pinhole camera as done in this 17th century manuscript.
- ✓Understand FOV(Field of Vision) of camera.

Experience

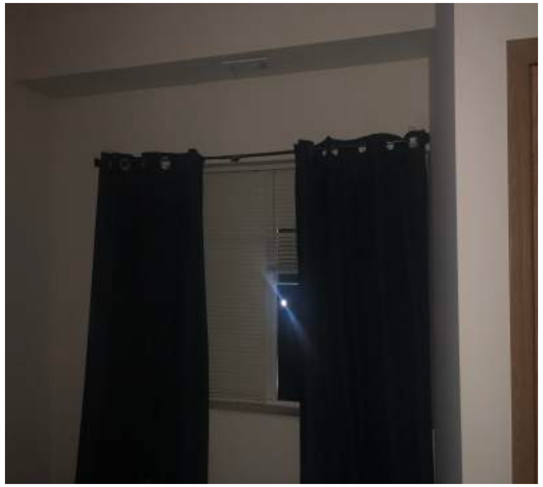
- This was a fun practical experiment to do. I was amazed to see the clarity of projected picture in the dark room and the effect of pinhole on the quality of it.

Assignment 1: Camera Obscura Set Up

Camera obscura setup covered windows

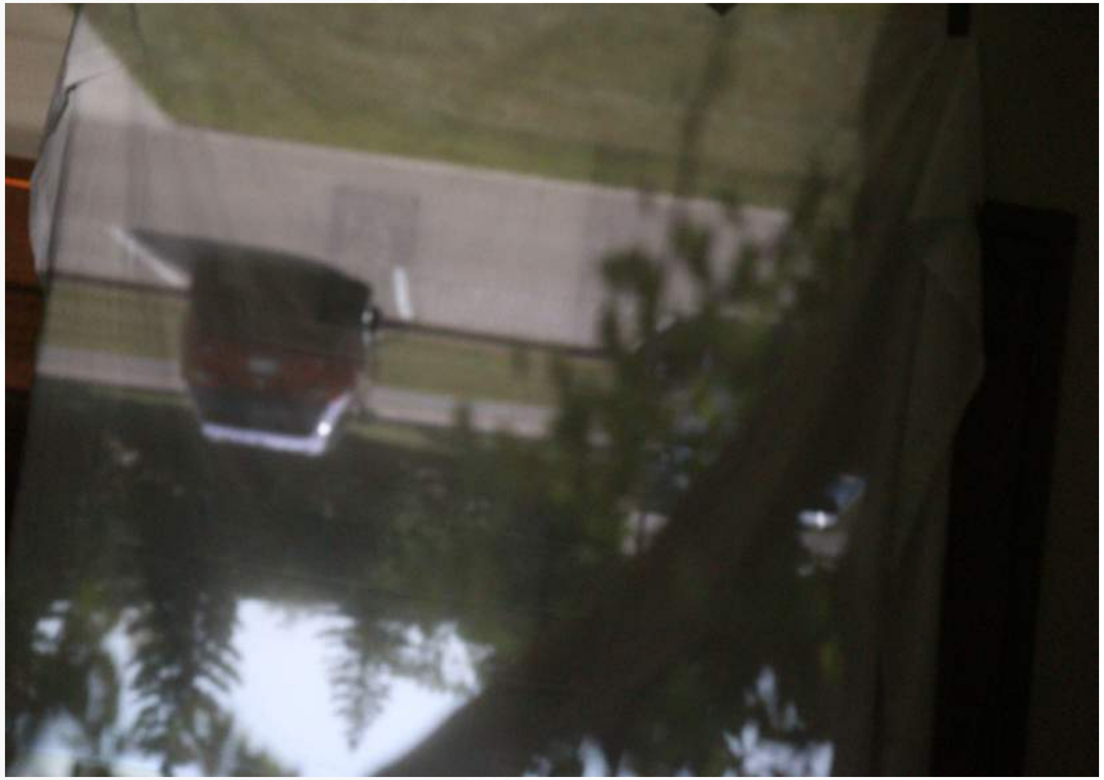


Pinhole(s) setup with camera position



Screen setup: wall covered with white sheet

Assignment 1: Camera Obscura result

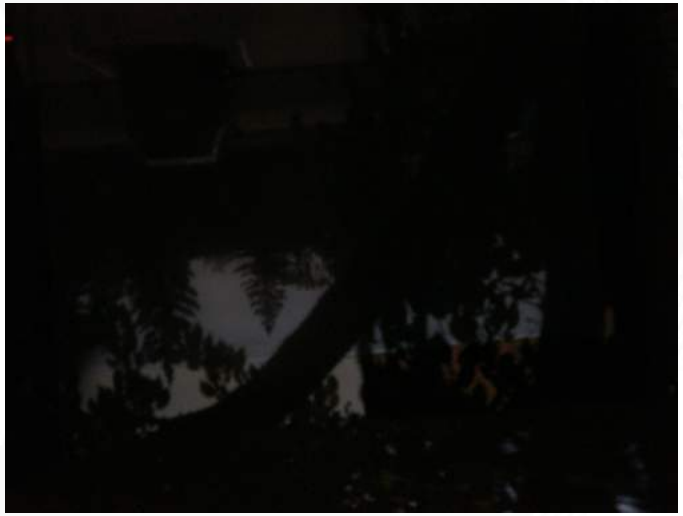


Obscura result, unedited

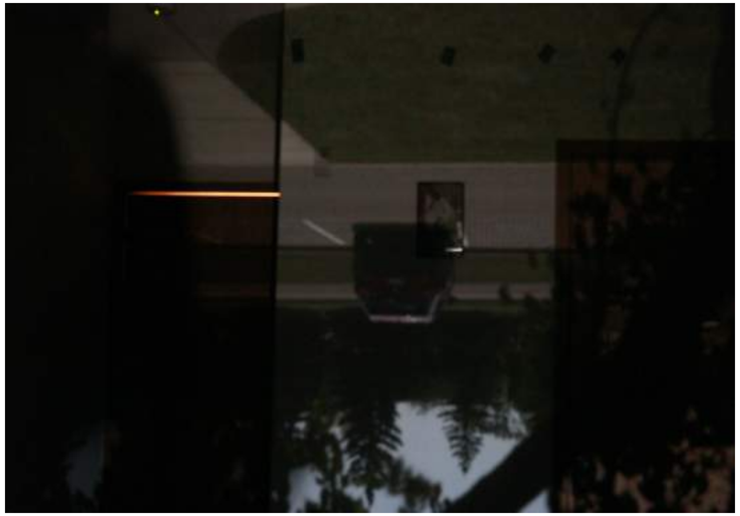
Actual Scene for camera obscura

Assignment 1: Camera Obscura results

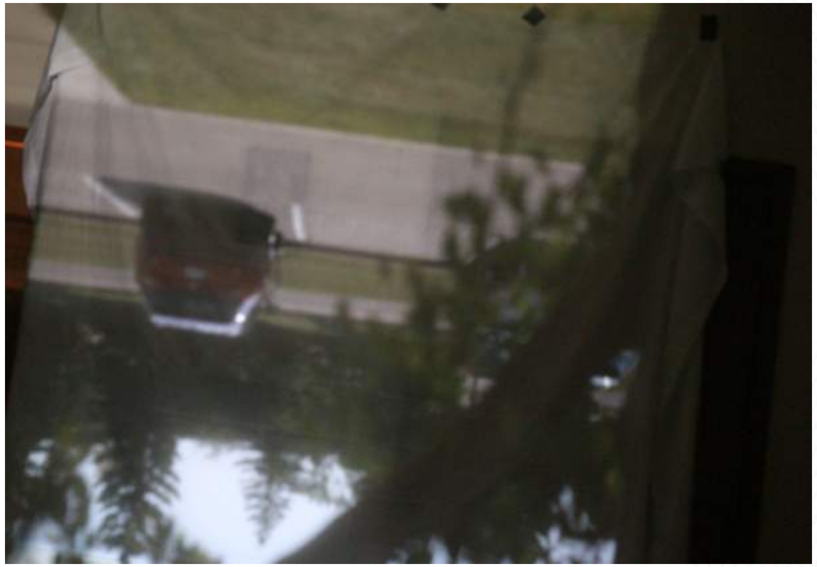
The projected images when different pinhole dimensions are used.



1/12”



1/8”



1/2”

Assignment 1: Camera Obscura

Challenges:

1. Finding an unobstructed view to capture.
2. Creating absolute darkness in the room by covering all the windows and light sources.
3. Hanging a white sheet on the wall to project the image.
4. Experimenting with long exposure timing for capturing the final scene.
5. Unable to capture long exposure picture with smart phone camera.

Assignment 2: Pyramid Blending

Goal:

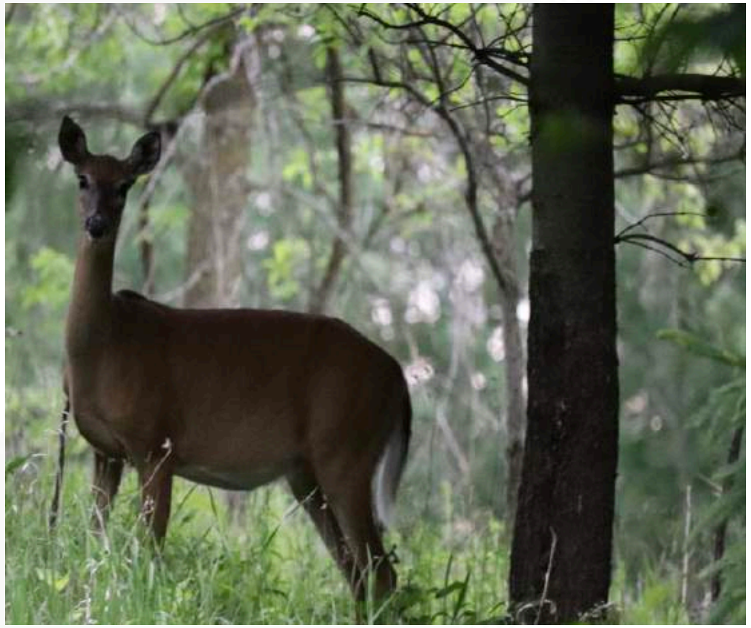
Blend two images by using a mask and a Laplacian Pyramid to blend each frequency independently.

Experience:

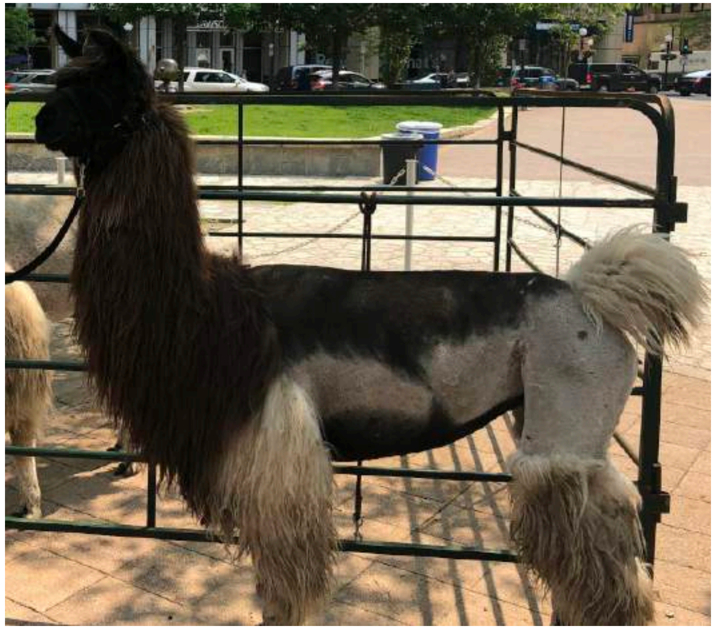
- It was fun to know and try different blending methods.
- I learnt using GIMP to create masks manually.
- Learnt using different pyramid blending to merge two images.
- Understand the importance of mask creation using the code.

Assignment 2: Pyramid Blending result

Source Image:

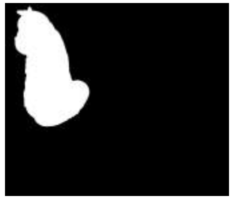


Deer

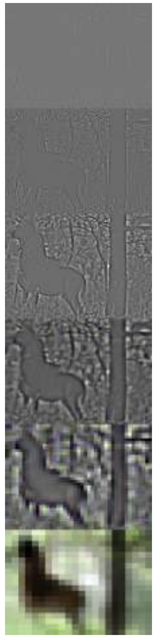


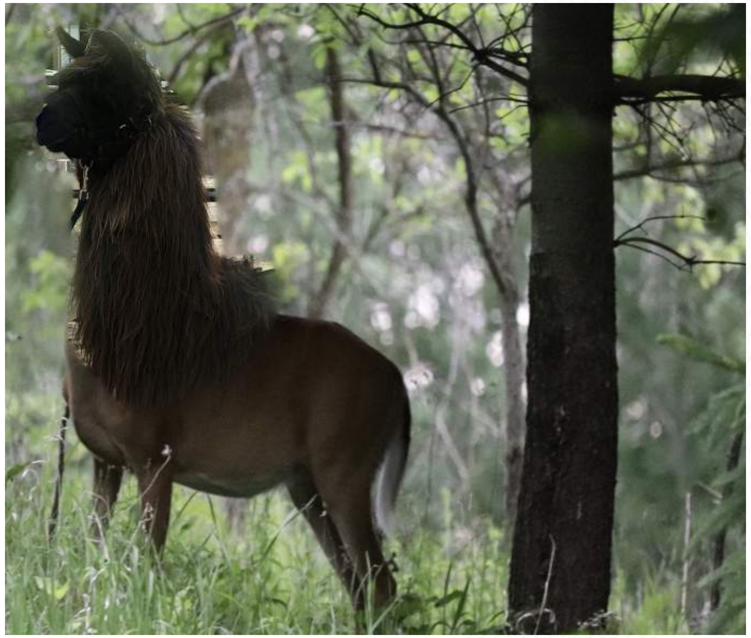
Llama

Mask used



Output Pyramid





Blended result

Assignment 3: Panoramas

Goal:

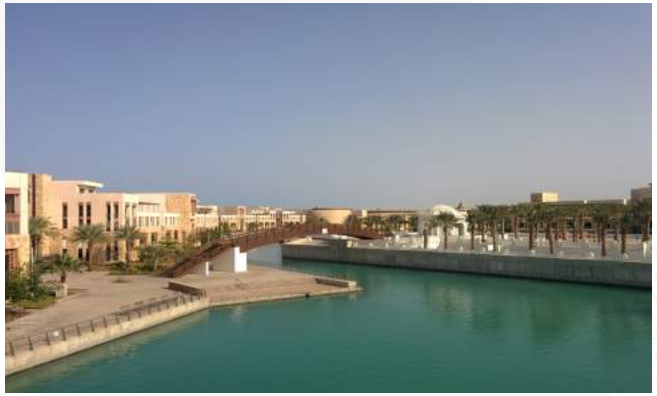
Create a panorama using multiple source images by matching features, computing homography, warping and aligning each image to form a smooth **blended** result.

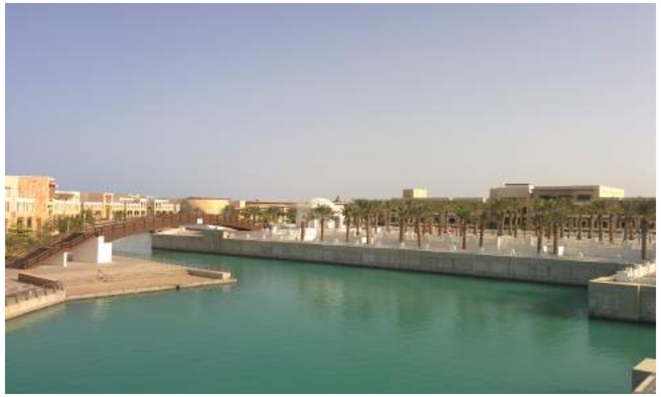
Experience:

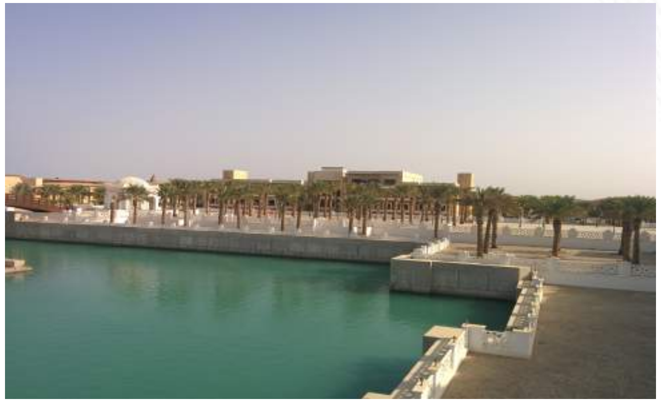
- ➢I really liked the whole process of generating a panorama.
- ➢Before doing this project, I didn’t understand the way google photos or phones generating panorama from multiple images. This project help me built that understanding.
- ➢I also learnt about feature detection, warping and image stitching in addition to alpha blending of images.
- ➢Automating the process of mask creation using distance transform and detecting energy in the image were some cool concepts that I understood.
- ➢Computing suitable alpha weights using distance masks(left and right).

Assignment 3: Panoramas (Sample Images)

Sample Input Images (provided in assignment):

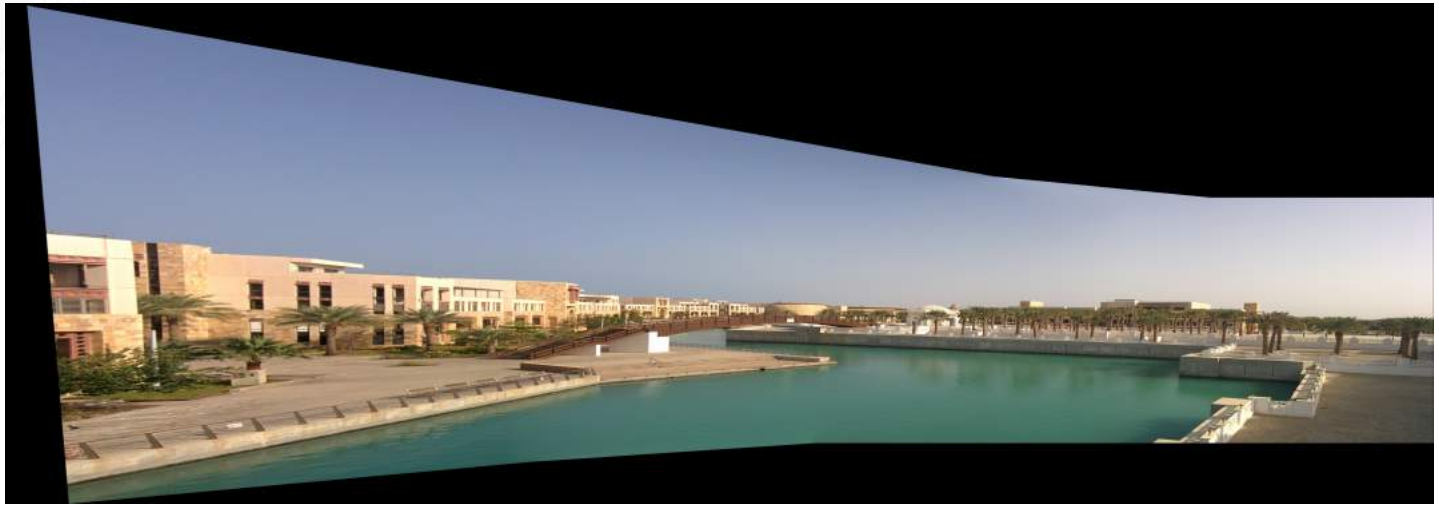






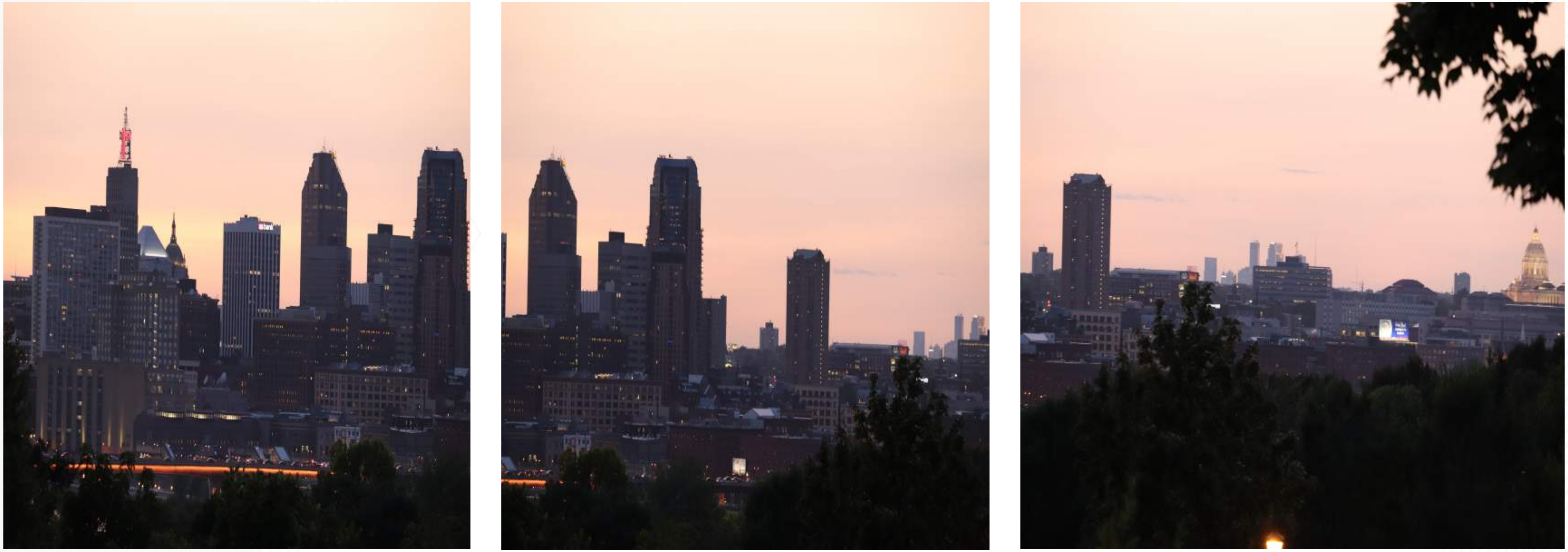
Output:

Alpha blended



Assignment 3: Panoramas (My Images)

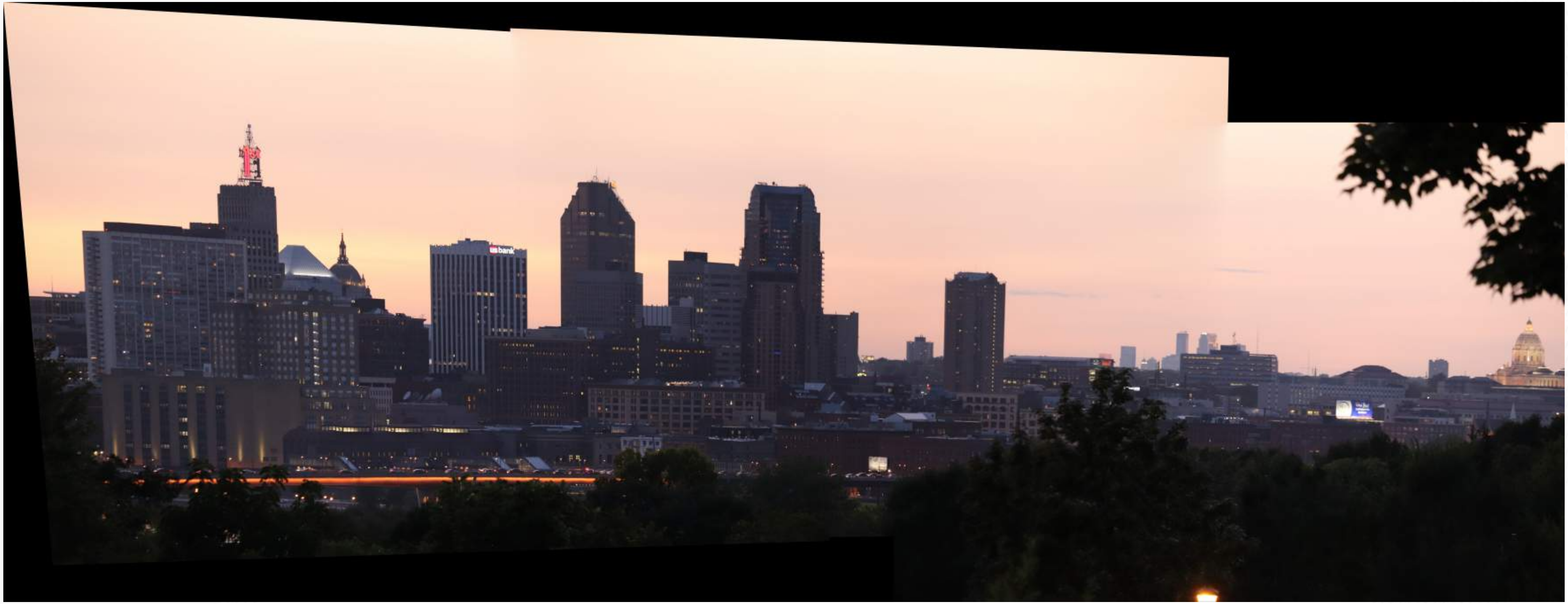
My Input Images:



Saint Paul, MN, Downtown pictures taken at golden hour

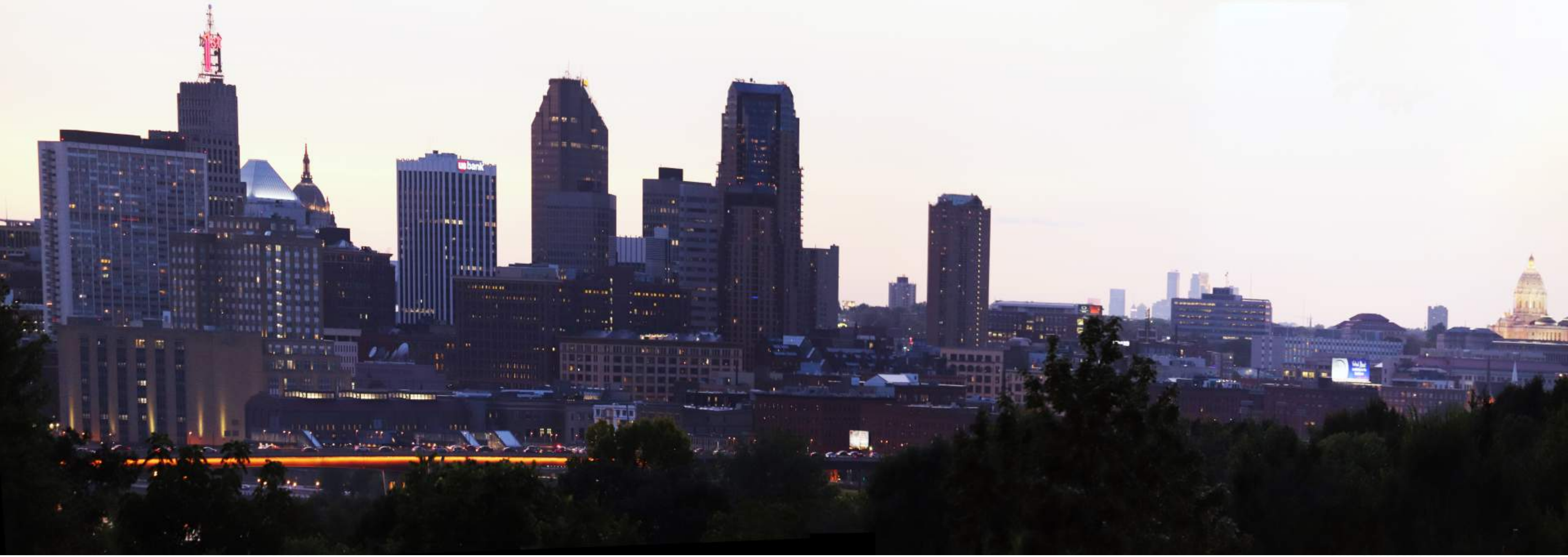
Assignment 3: Panoramas (My Images)

Result Image after *alpha blending*:



Assignment 3: Panoramas (My Images)

Result Image after edit in photoshop:



Assignment 4: HDR

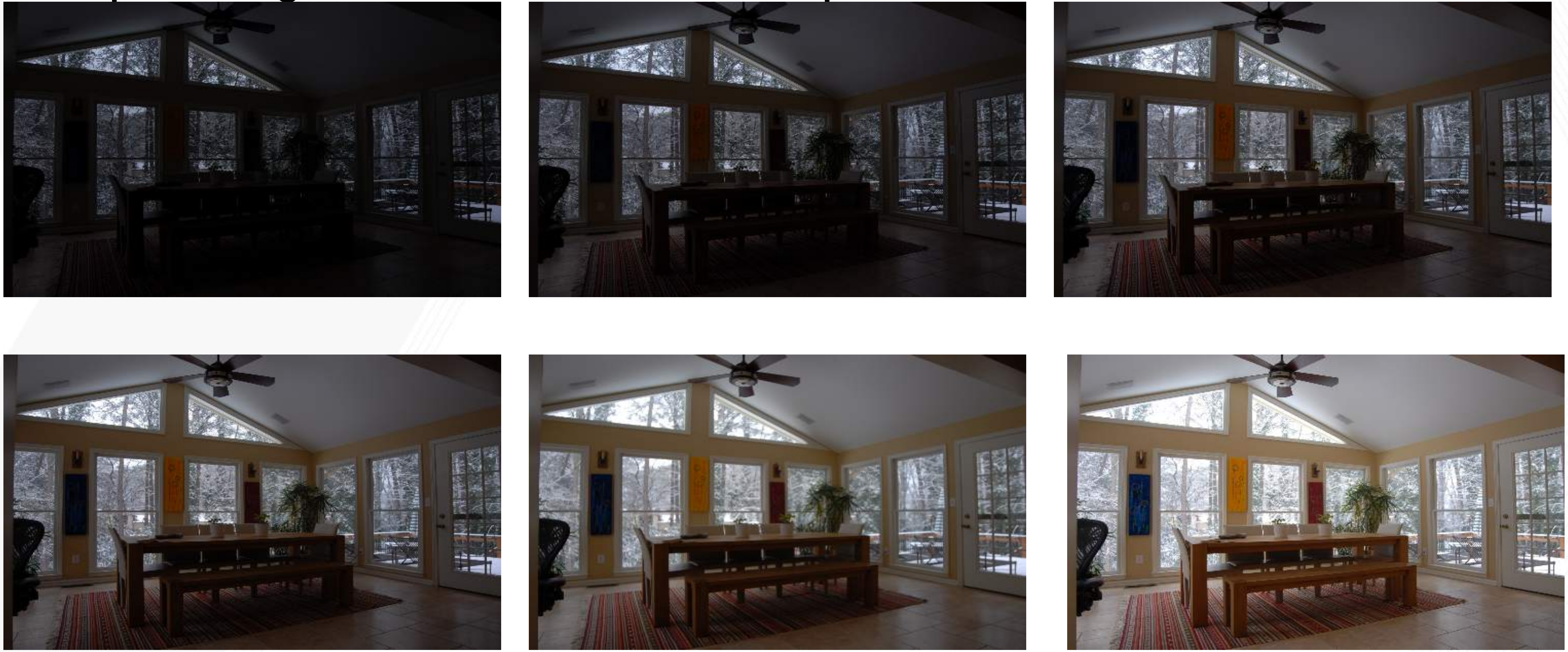
Goal: To create a HDR image using multiple images of the same scene with different exposures capturing the original scene radiance.

Experience:

- Got good exposure to the High Dynamic range(HDR) spectrum and photos.
- Learnt different ways of capturing the light intensities present in the scene.
- Developed the understanding to capture the radiance using multiple images of the same scene, how to do so, what things to control while capturing HDR.
- Learnt about different tone mapping methods like Durand, Drago, Reinhard *etc.* to bring out the radiance in the image as seen with naked eyes.
- Comparing my results with the output of photographer **Klaus Herrmann** was very interesting.

Assignment 4: HDR (Sample Images)

Sample images taken at different exposures



Assignment 4: HDR (Sample Images)

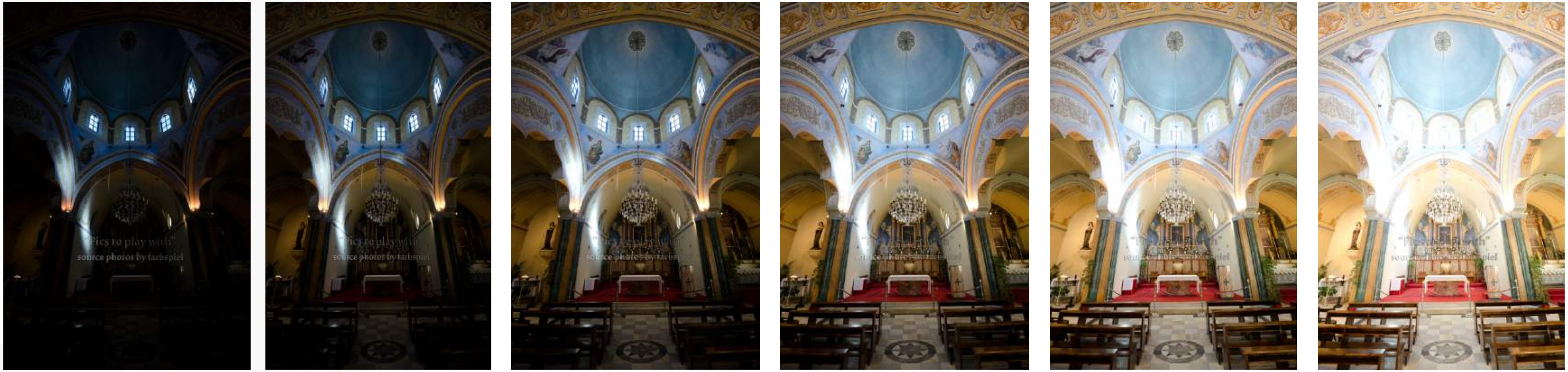
Sample image: Basic HDR result

Sample images: Best HDR result

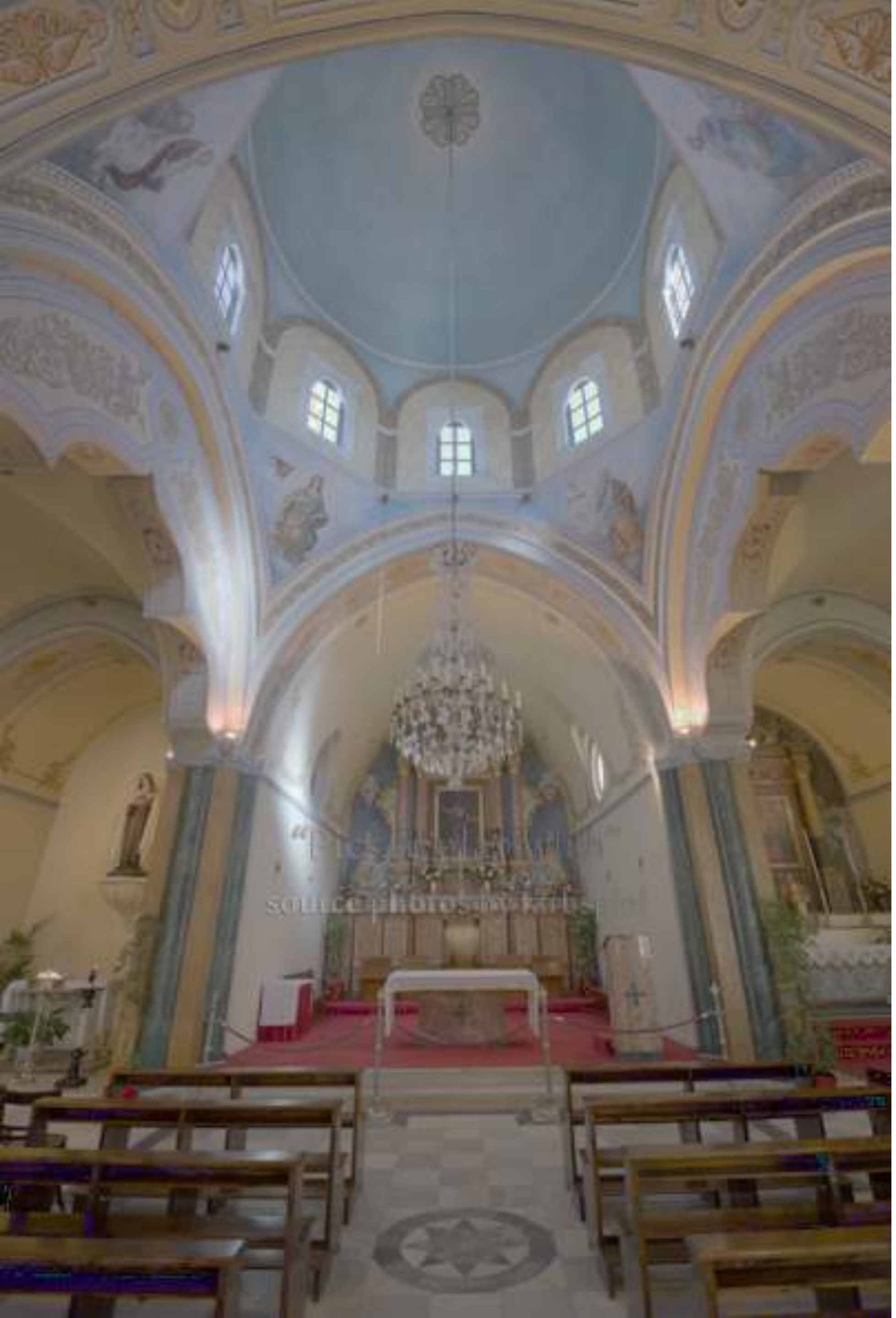
Assignment 4: HDR

Input Image Set:

Santorini Cathedral



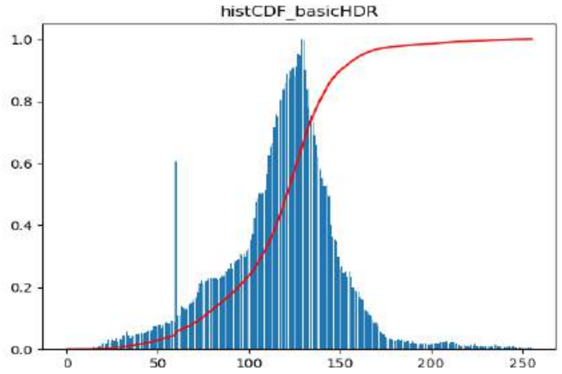
- Camera Nikon D7000, Lens: Nikon AF-S DX Nikkor 10-24mm f/3.5-4.5G ED (at 10mm)
- Six exposures: +2EV to -3EV in steps of 1EV
- f/3,5
- 1/20s – 1/640s
- ISO 800
- Hand-held (autobracketed)



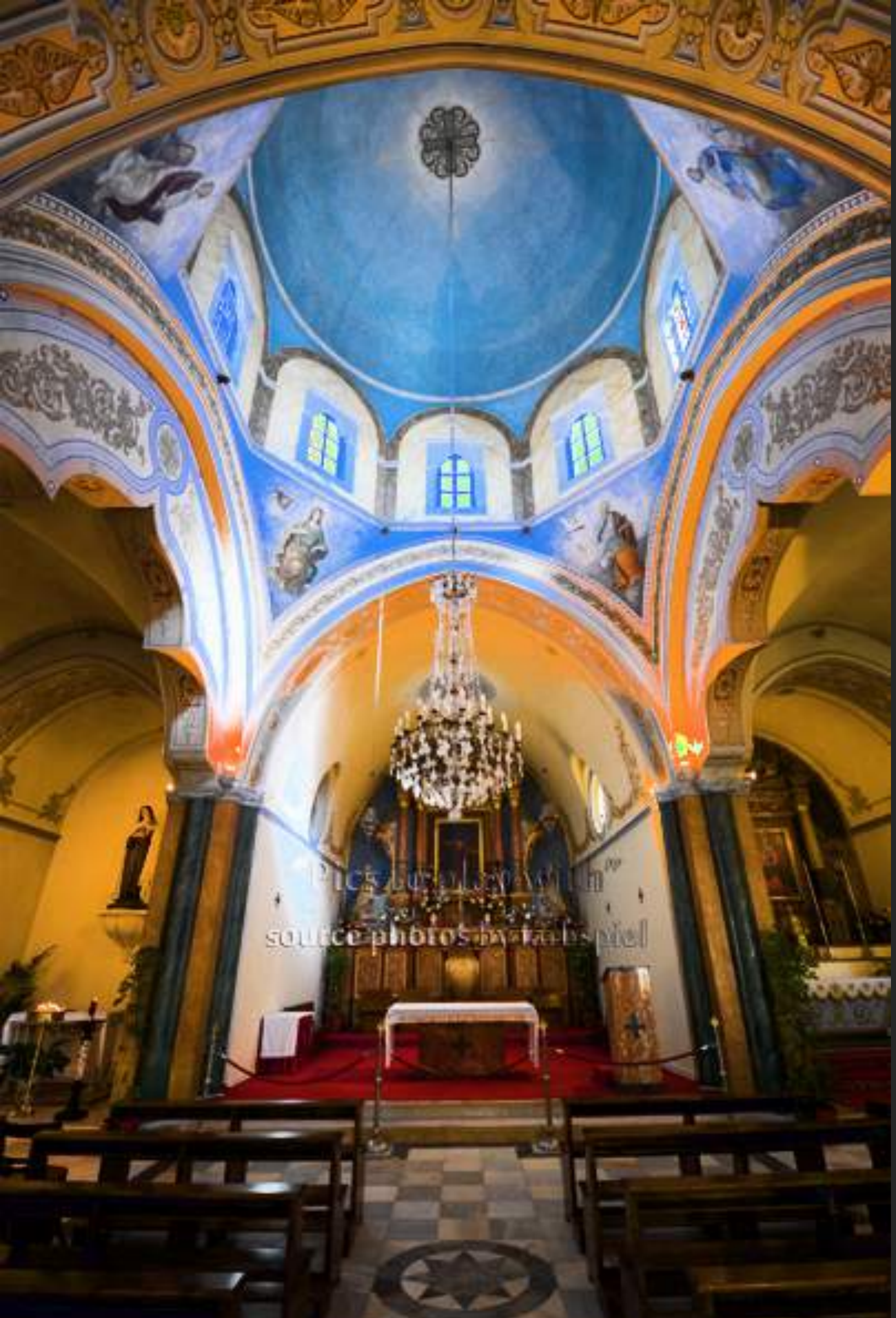
Assignment 4: HDR Basic Result

- Basic HDR Created using Input Images and calculating Radiance Map.

- Histogram plot basic HDR:



Source images copyright by Klaus Herrmann – farbspiel-photo.com – all rights reserved. Go to farbspiel-photo.com/ppw for details.

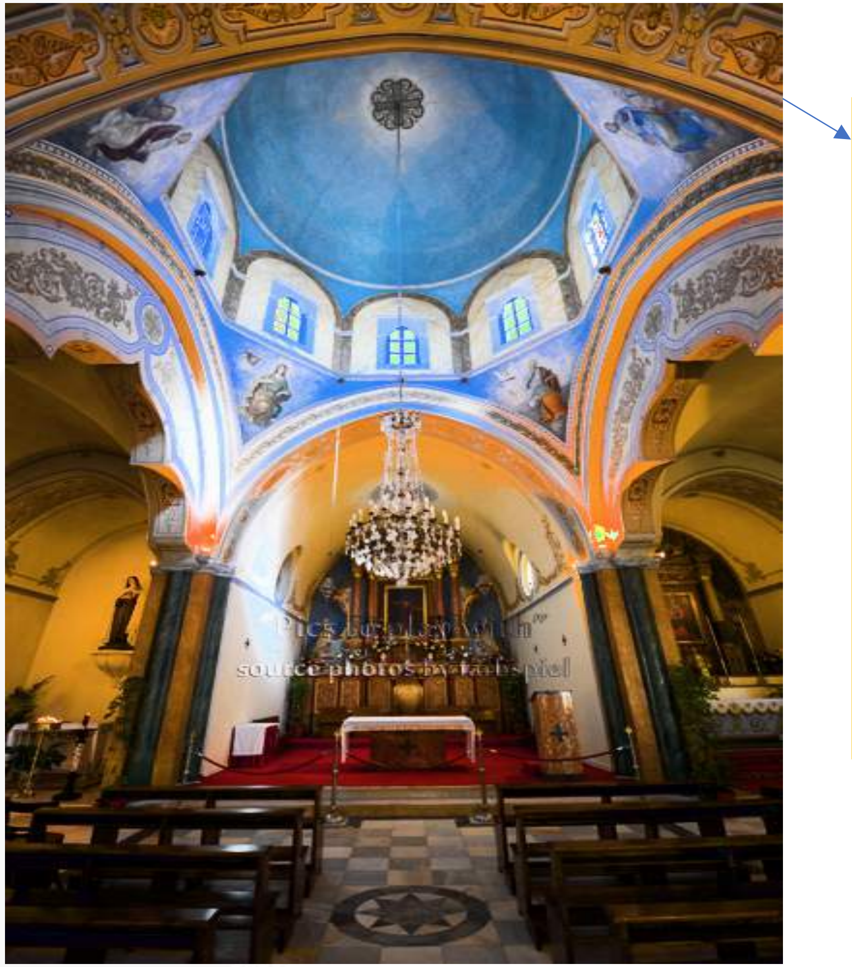


Assignment 4: Best HDR Result

Best HDR created by using Histogram Equalization and Adaptive Gamma Correction With Weighting Distribution [4] on Basic HDR result

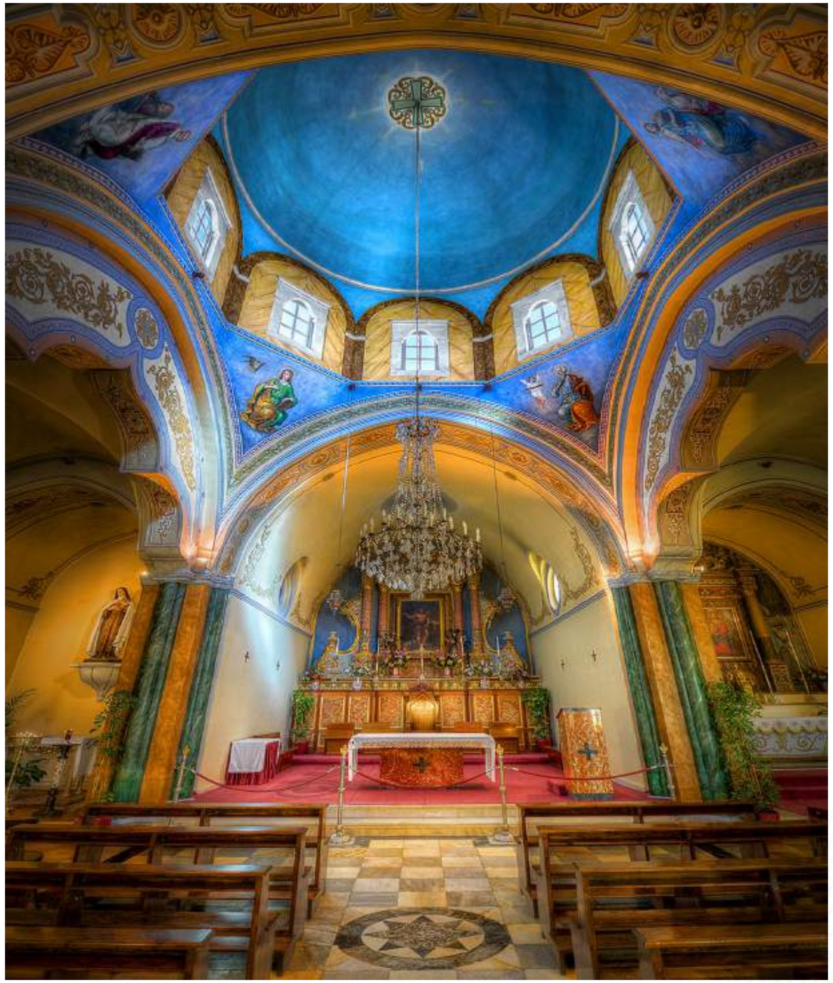
Assignment 4: Comparison HDR result

My Result Best HDR



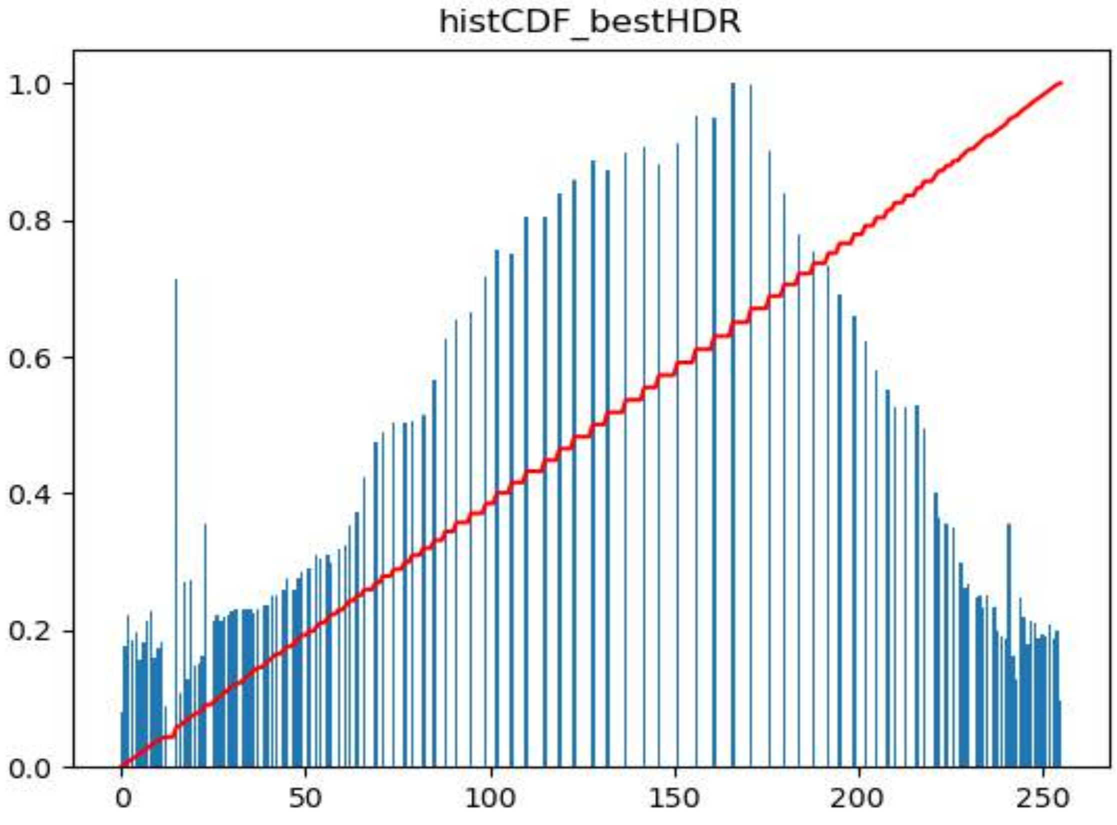
Similar colors as right, overexposed walls and windows on top,
Little dark floor and benches,
Better than Basic HDR

[*Klaus Herrmann's result*]()

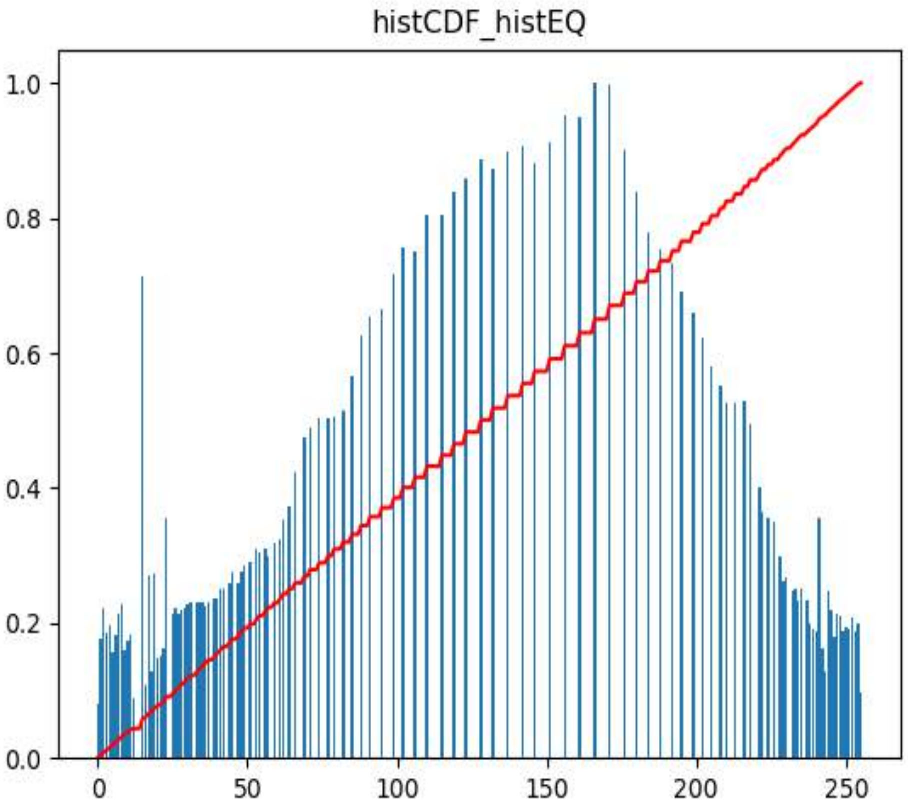


More vibrant Colors, no over exposure,
Better light, no darkness

Assignment 4: Best HDR Histogram



Best HDR Histogram after equalization



Histogram Equalized and CDF

Assignment 5: Video Textures

Goal:

To identify a part of a video which can generate an infinite unending video with:

- Smooth frame transition.
- compute the similarity matrix between frames of a video.
- Identify similar frames and use them to create an unending gif.
- Find the biggest loop available using suitable alpha to get the loop.

Experience:

- First assignment working with video was fun.
- Learnt about the frame extraction and video creation using ‘ffmpeg’ .
- Finding the correct alpha for seamless gif was time consuming and challenging.
- Getting a suitable video where the background is stationary was difficult too.

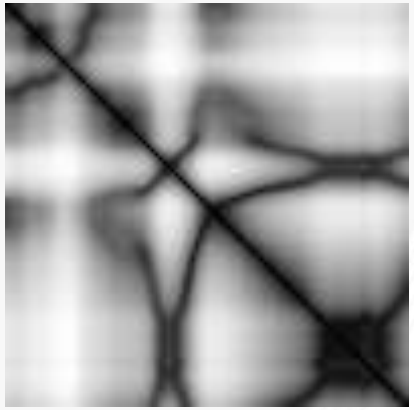
Output Resources can be accessed here at One drive.

Assignment 5: Video Textures

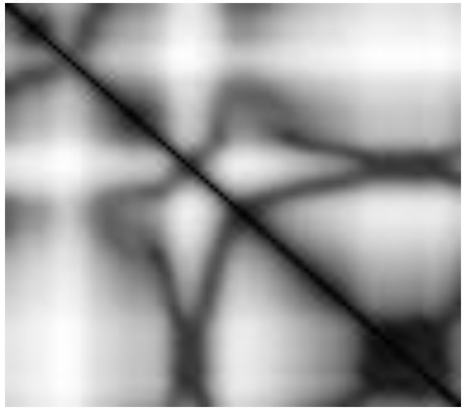
Best Result Sample Input: alpha used 0.02

Frame 000

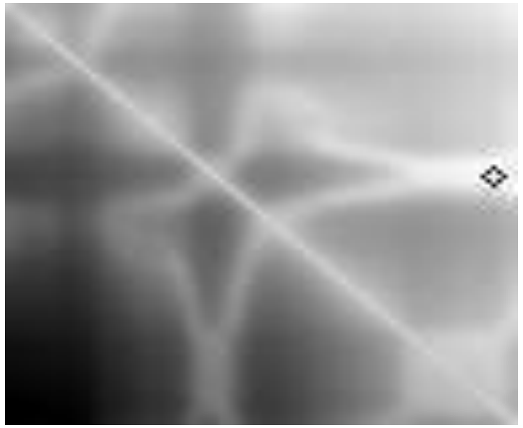
frame 0052



candle_diff1: Similarity



candle_diff2: Transition

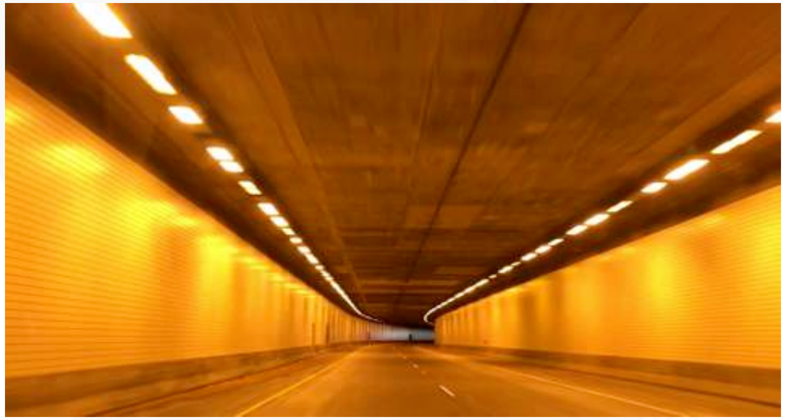


candle_diff3: Scoring

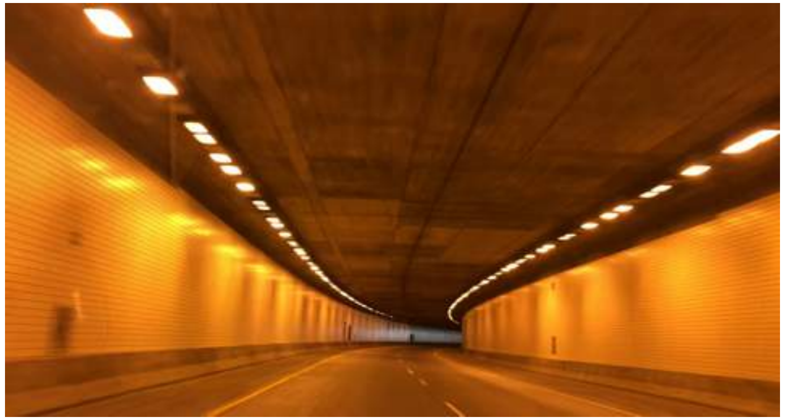
Transition Matrics for Sample Candle input

Assignment 5: Video Textures

Best Result Sample Input: alpha used 0. 0148

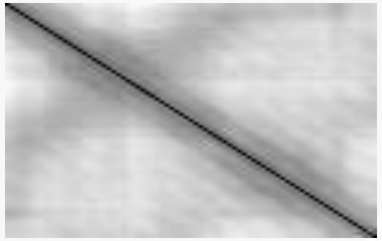


Frame indx 2

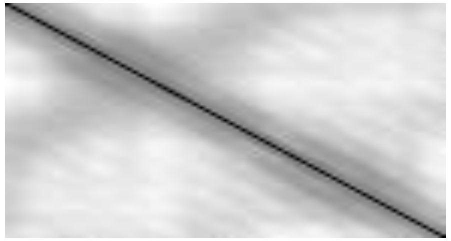


Frame indx 70

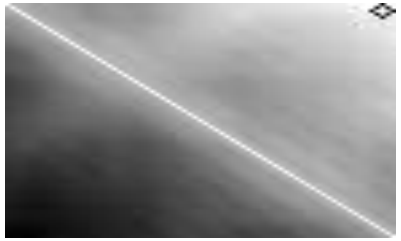
Output Resources can be accessed here.



- Image: Diff1.png
- Method: computeSimilarityMetric



- Image: Diff2.png
- Method: transitionDiffrence()



- Image: Diff3.png
- Method: findBiggestLoop()

Best Score location (2, 70)

Midterm Project

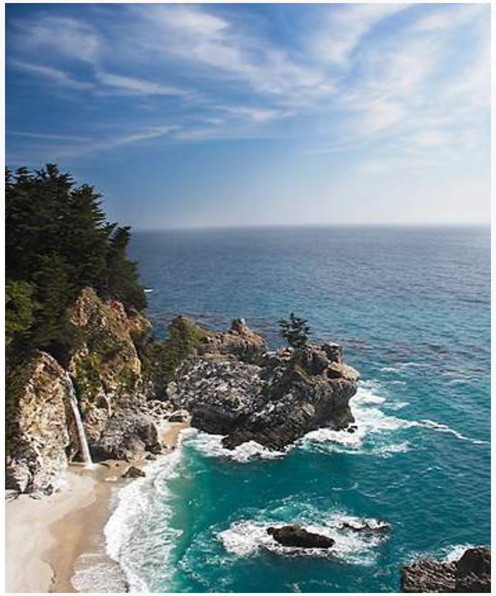
Goal:

The goal of this project was to replicate results of the seam carving paper[2] [3] on seam carving.

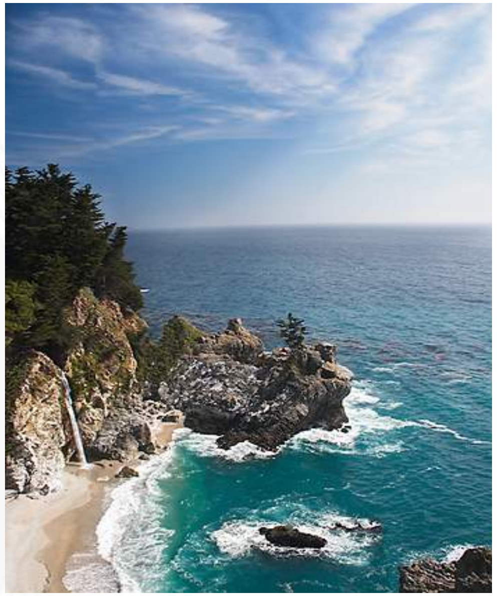
Experience:

- This was the most difficult and time consuming compared to other assignments and projects in this class.
- Learning how to make assumptions when the implementation details are left out was one of the challenges for this project. The paper had so many details hidden and vague that we needed to make assumptions and try out different ways to achieve the result.
- I was able to finish the project, but my results were not satisfactory when compared to the paper.
- Creating the project report in latex was another learning curve for me.
- The project helped build the understanding of how to resize image keeping in mind image content but was tough.
- Introduction to dynamic programming was interesting and new for me.

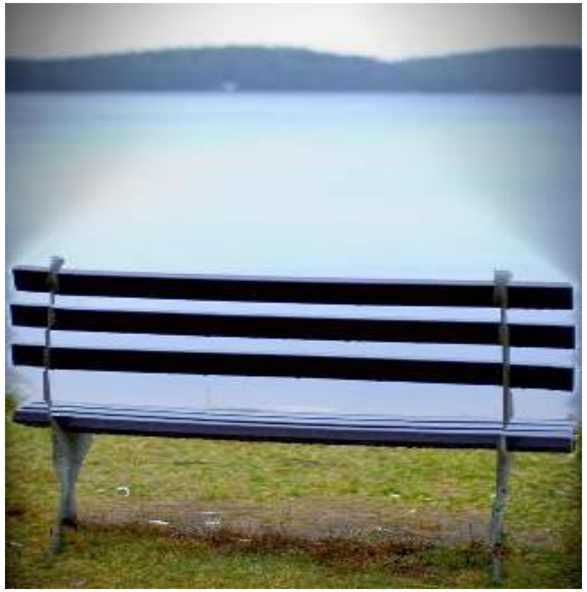
Midterm Project: Results



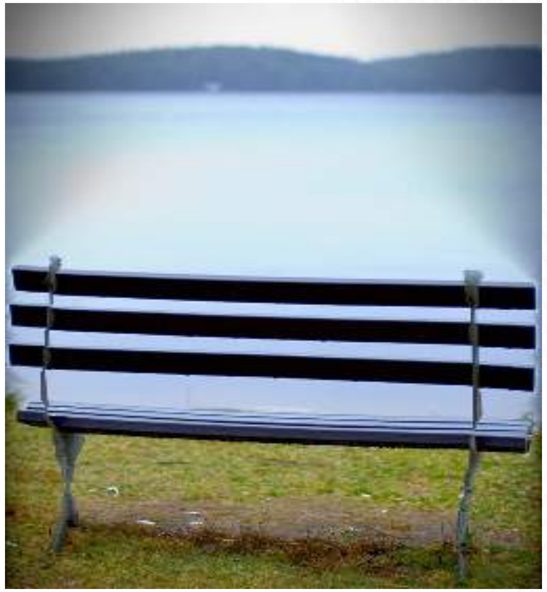
Waterfall 2007 paper



Waterfall My output



Bench, author’s result



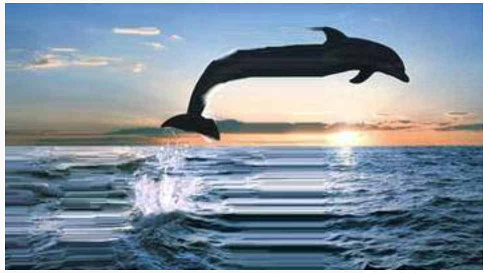
Bench: My output

Seam removal, 50% reduction for the waterfall image using backward energy.

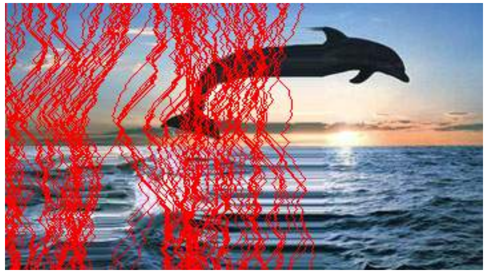
Seam removal for the bench image using backward energy to reduce the size by 50%.

Midterm Project: Results

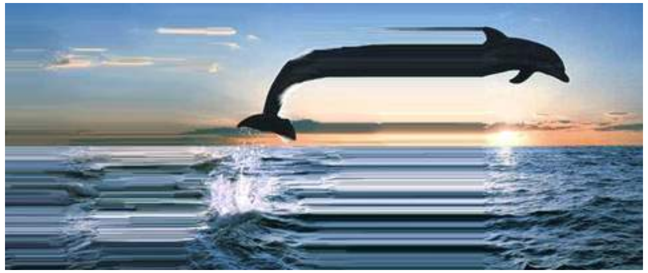
Increase dolphin size by 2 step 50% increase using seam insertion algorithm



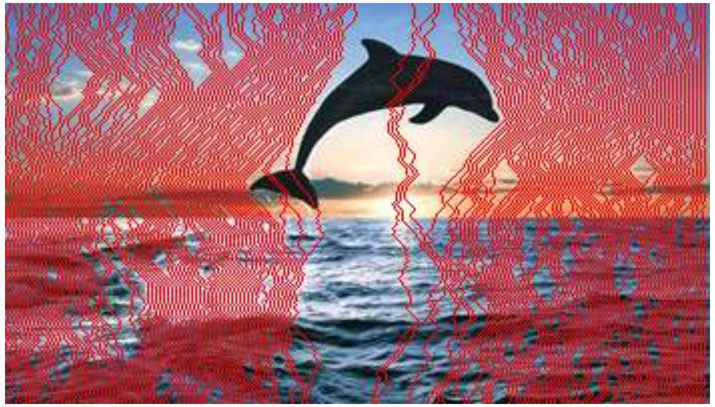
One step 50% Seam insertion



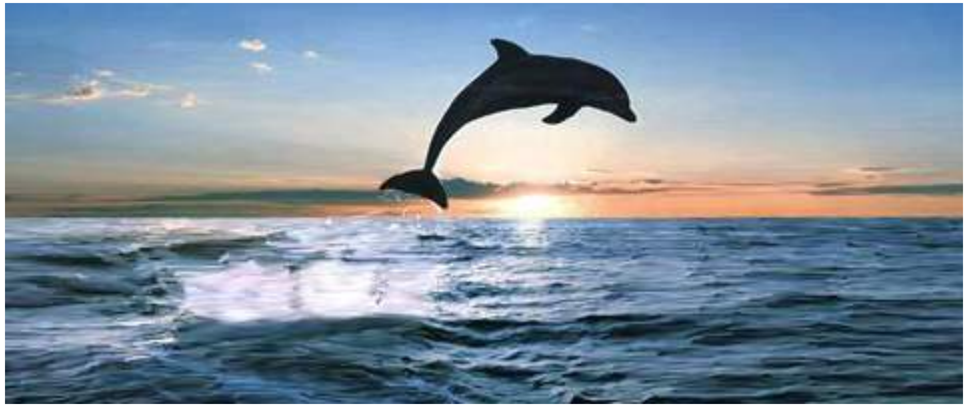
2nd step 50% Seams mapped for insertion



Two step 50% Seam insertion result



One step 50% Seam insertion by authors

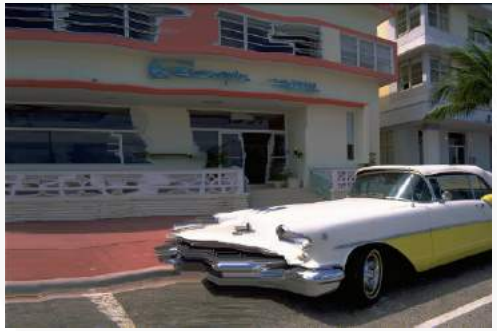


Increase size Dolphin 50% and 2 step 50 % increase: author's result

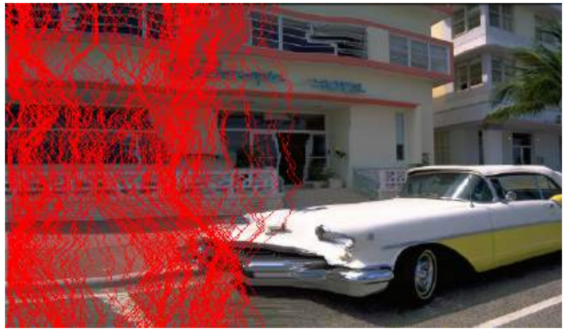
Georgia Tech
CREATING THE NEXT

Midterm Project: Results

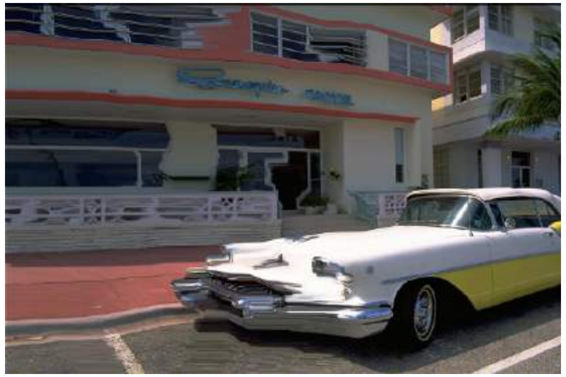
Expanding Car Image by Seam insertion Back ward and forward energy. Problems while seam insertion in my results



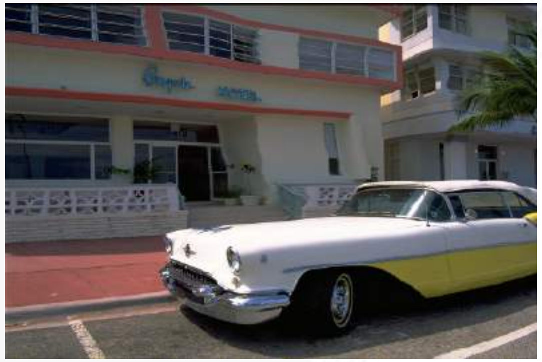
Backward Energy Seam insertion: My result



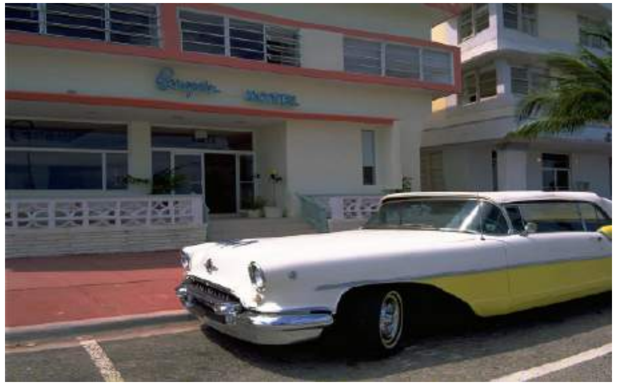
Backward Energy Seam insertion map



forward Energy Seam insertion: My result



Backward Energy Seam insertion: Author's output



Forward Energy Seam insertion: Author's output

Final Project

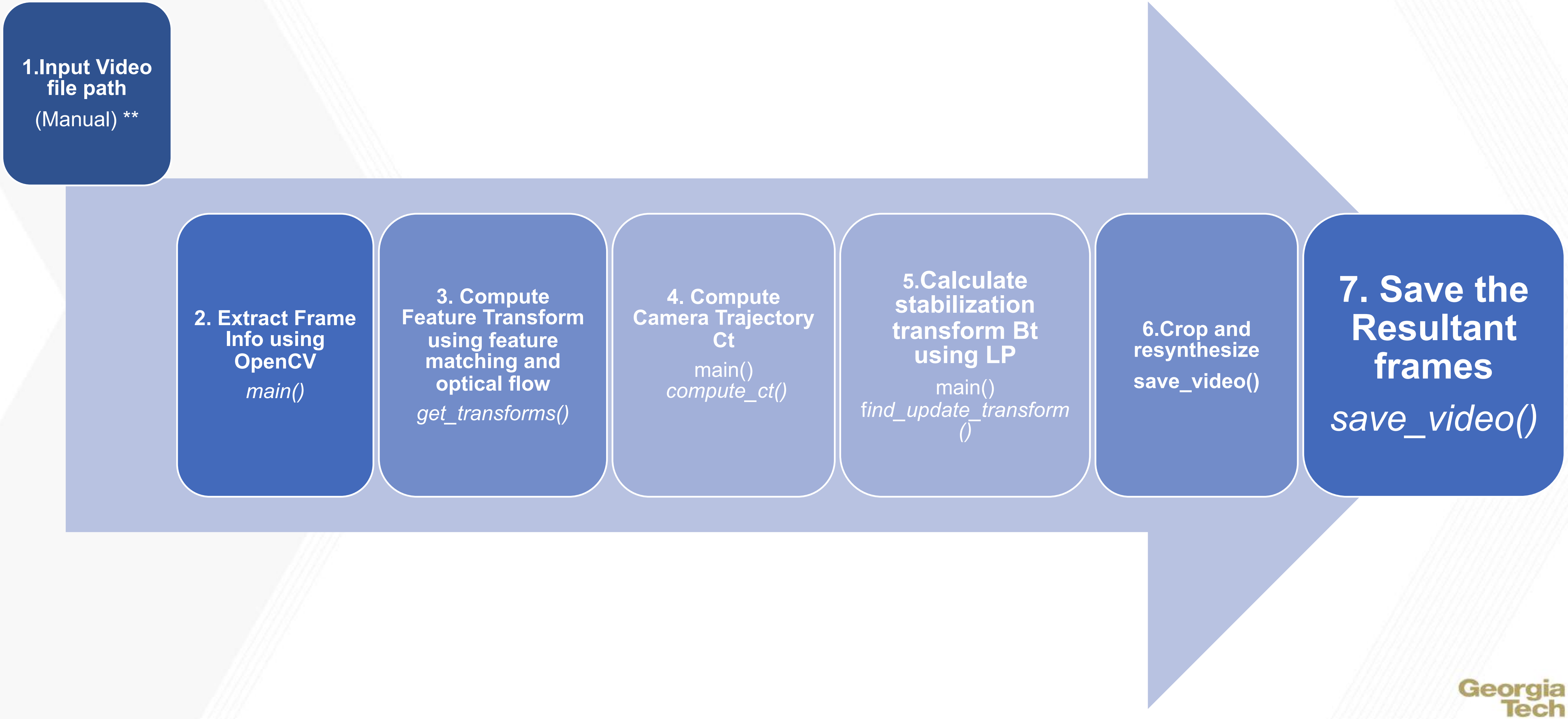
Goal:

To generate a stabilized video by removing undesired motion from the original shaky, casually shot video clip.

Experience:

- This project was very interesting. The algorithm described in paper[5] is used in YouTube for stabilizing videos.
- Professor Essa is one of the co-authors on the paper[5] and one of the lecture module covers this topic so it was an easy decision to pick this paper.
- The project was challenging, and it involved doing Linear programming which was a new zone for me.
- I understood the ways to make the video stable by computing Camera path for original video. To some extent the results can also be achieved by applying average of the camera movement in x an y plane and generating a new affine which can be applied to the frame for generating stable video.
- Authors have used L1 robust stabilization in the paper.
- The project doesn't implement Saliency constraints and Rolling Shutter/Wobble suppression due to time constraints.

Final Project : Pipeline



1.Input Video file path

(Manual) **

2. Extract Frame Info using OpenCV

main()

3. Compute Feature Transform using feature matching and optical flow

get_transforms()

4. Compute Camera Trajectory Ct

main()
compute_ct()

5.Calculate stabilization transform Bt using LP

main()
find_update_transform()

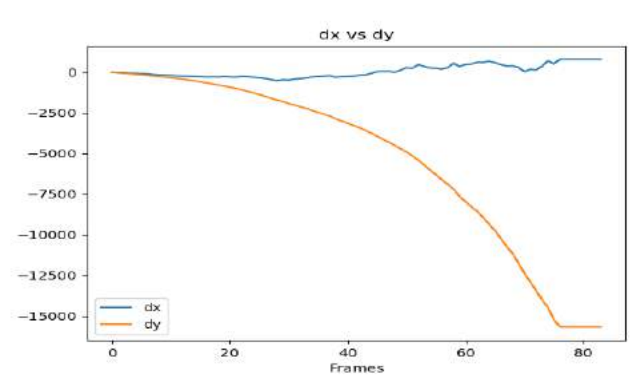
6.Crop and resynthesize

save_video()

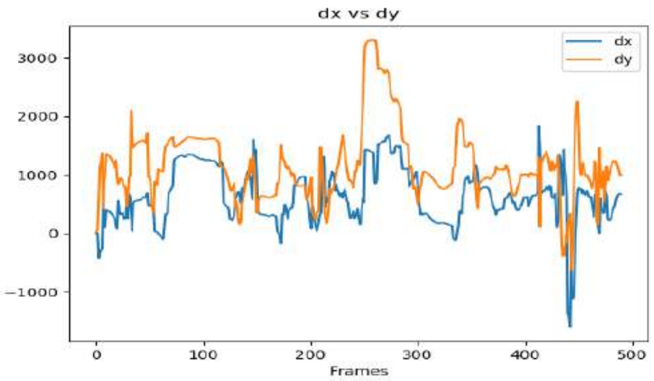
7. Save the Resultant frames

save_video()

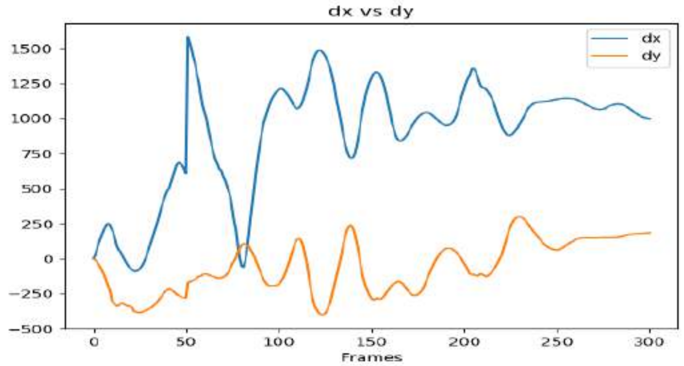
Final Project : Camera Trajectory Plots



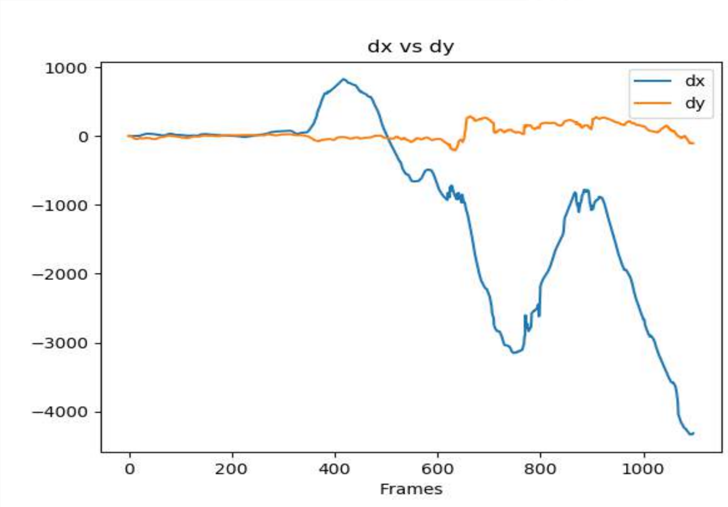
Input_tunnel.MOV camera path



Input_fidget_spinner.mp4 camera path



Wind chime camera path



dx and dy plot for camera path Ct for skater video, 1078 frames

Output results are available here at one drive

Final Project : Libraries Used

Libraries used:

The following libraries are used to implement the code:

1. **Numpy**: for doing numerical operations using multi dimensional array and matrices.
2. **OpenCV Python**: for reading and saving video frames.
3. **matplotlib.pyplot**: used to create a figure, a plotting area in a figure, plot some lines in a plotting area, decorate the plot with labels. Mostly used here for plotting camera path and stabilized camera path.
4. **Pulp**: to work with mathematical problems and linear programming/optimization by adding multiple constraints as described in the research paper.
5. FFMPEG: for working with Video codecs and creating video/gif from frames.

References:

1. A Multiresolution Spline With Application to Image Mosaics.
2. Avidan, S., & Shamir, A. (2007). Seam Carving for Content-Aware Image Resizing. *ACM Transactions on Graphics (Proc. SIGGRAPH),* 26(3), (pp. 10-es).
3. Rubinstein, M., Shamir, A., & Avidan, S. (2008). Improved Seam Carving for Video Retargeting.
4. Efficient Contrast Enhancement Using Adaptive Gamma Correction With Weighting Distribution Huang et al.
5. Grundmann, M., Kwatra, V., & Essa, I. (2011). Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths.
6. OpenCV Python documentation : https://docs.opencv.org/3.0-alpha/doc/py_tutorials/py_tutorials.html
7. Class Lectures CS 6475 Fall 2020.
8. Piazza Class Forum CS6475: Fall 2020