# CS-766

# Assignment-2: Image Panorama

**Rosaleena Mohanty**

**Anmol Mohanty**

# CS-766

# Assignment-2: Image Panorama

**Rosaleena Mohanty**

**Anmol Mohanty**

**TABLE OF CONTENTS**

**CS-766**

**Assignment-2: Image Panorama**

**Rosaleena Mohanty**

**Anmol Mohanty**

## 1. Project Description

Forming a panoramic mosaic involves combination of multiple images where any two consecutive images share a common portion of field of view resulting in a final image that spans a wide angle view of any scene that normally cannot be captured by a single shot using a camera. The best panoramas are the ones that have almost exact overlapping portions between the input images and possess similar, if not identical, exposures leading to seamless blending.

Here, in this project, we try to create such mosaics by capturing multiple images of the scene, running the algorithm to stitch and blend input images using MATLAB.

## 2. Input Datasets

We used the following camera setup for image acqusition:
Camera: Canon SX100 IS tag 6418203978
Rotator: Kaidan quick pan, Kaidan camera bracket
Resolution: 480x640
Focal length: 682.05069
k1 = -0.22892, k2 = 0.27797

We acquired two datasets: (i) Indoor: Madison State Capitol: view of the ceiling
(ii) Outdoor: Parking lot over-looking WIMR building

## 3. Implementation Details

The basic steps involved in the algorithm for creating the panorama are:

a. Image acquisition
b. Cylindrical coordinate projection
c. Image pair alignment, finding corresponding feature points, transformation and merging.

d. Blending, Stitching and cropping.

**4. The core Panorama part:-**

High Level algorithm:-

**How does it work?**

The basic method works as follows:

- Capture two images.
- Convert images into cylindrical system(see alternatives and explanation below)
- Detect **SIFT** key points in both images and compute the set of matching points
- Apply **RANSAC** to estimate a **homography** that transforms the images such that the points coincide
- Transform the images using this homography.
- Blend the images together.
- Capture the next pair of images and repeat.

**a. Capturing Images:-**

The images were taken with a Canon SX100 IS tag 6418203978 camera, Kaidan head and tripod.

### b. Cylindrical Coordinates

Captured images were warped to cylindrical co-ordinates using the system below.

$$\Theta = \arctan(x / f);$$

$$h = y / \sqrt{(x^2 + f^2)};$$

In the above equations, f is the focal length, x is the x coordinate of the 2-d image representation, and y is the y coordinate of the 2-d image representation. This gave us a cylindrical coordinate for each of our x, y positions in our images. After this conversion, we warped our images back into the image plane coordinates using the following equations.

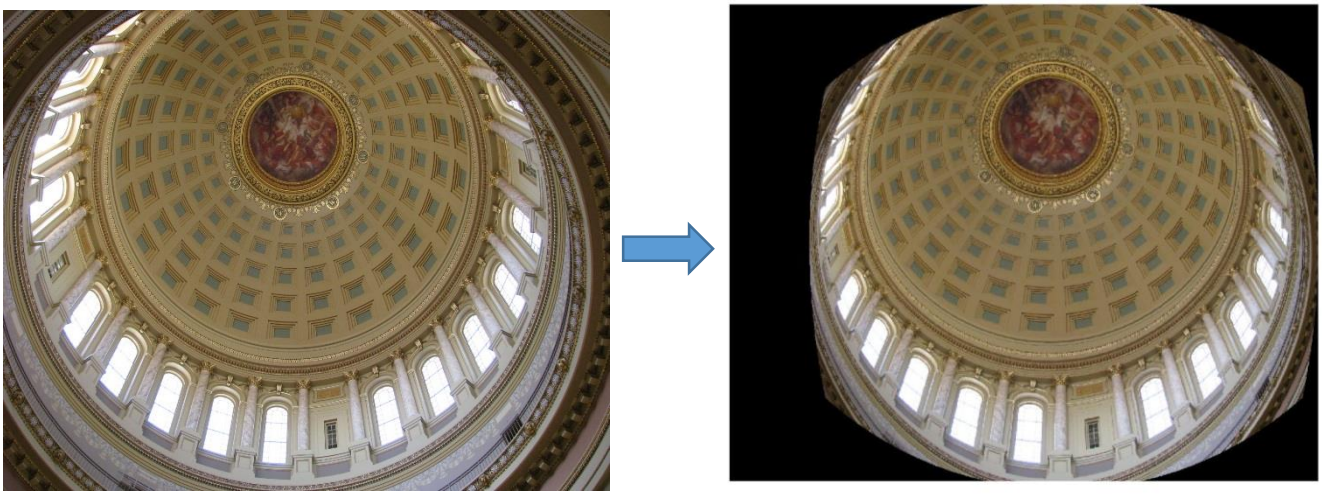
$$new\_x = s*f * theta + x\_center;$$

$$new\_y = s*f * h + y\_center;$$

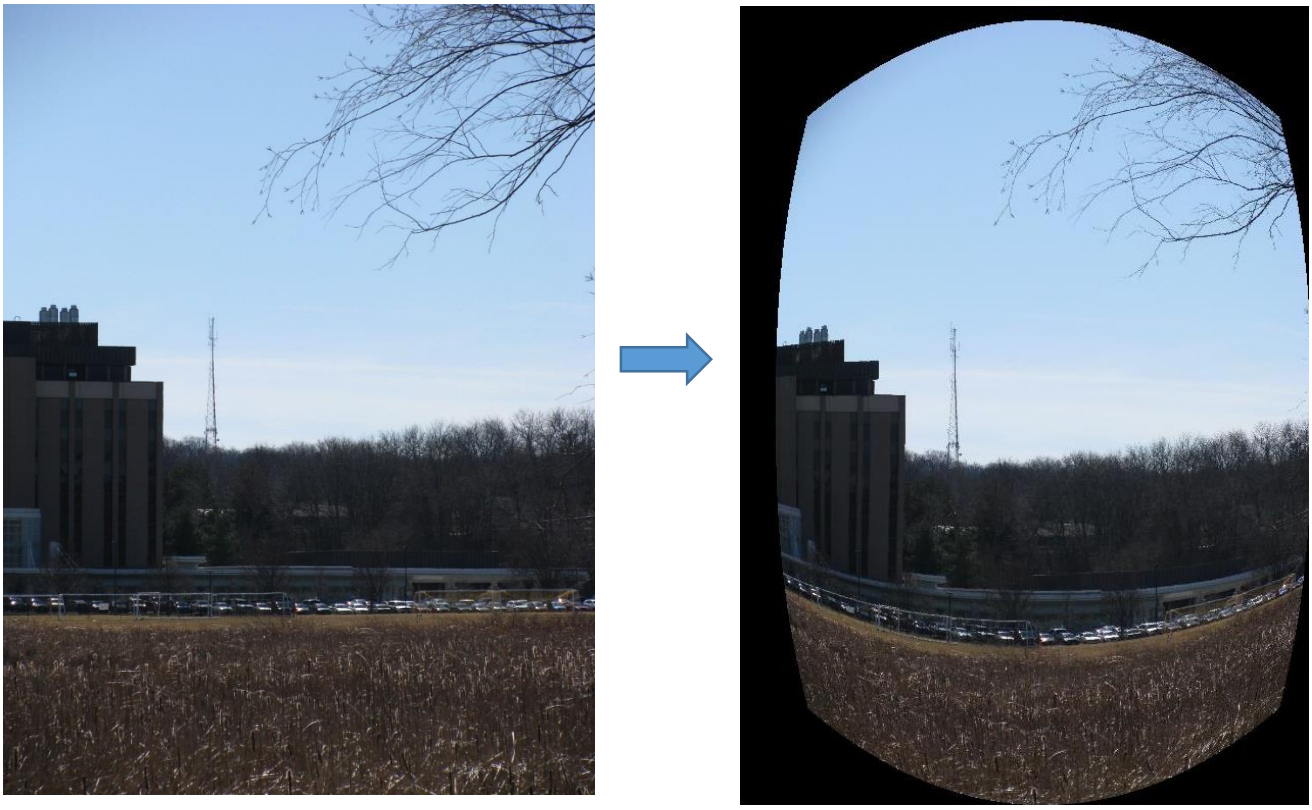
Please note: in this project we assumed the scaling factor s to be equal to 1.

In the above equations, the terms x_center and y_center represent the center of the unwrapped cylinder for the cylindrical coordinates. We used these new_x and new_y values to recreate our original images as cylindrical images. Below, we show our original image (left) and its cylindrical image representation (right).

Some of the outputs at the cylindrical stage:-



F=782



F=782

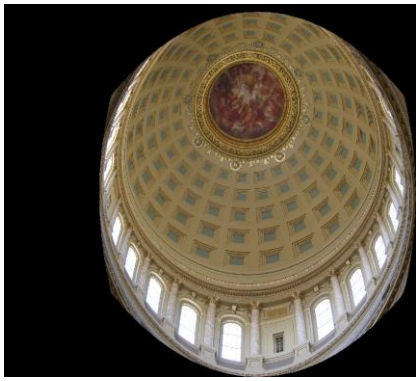
An important to note is the fact that we didn't have our camera zoomed all the way out and hence did not know the accurate focal length of the setup when we shot photos. We couldn't have it zoomed all the way out because of the nature of the scene under consideration wasn't permitting that. So we assumed a convenient value of the focal length to do the cylindrical warping. We also came up with a technique to apply SIFT on the captured images to find out the matching points and then use homography to project the images onto a common plane for stitching them together.

### c. Interesting observation on changing the focal length:-

The focal length of the camera changes upon using the optical zoom obviously. However an interesting thing to note is the fact that the focal length has a strictly increasing relationship with the zoom. And with increased focal length the output of cylindrical image becomes flatter which makes sense because the images are farther off necessitating the zoom and hence would appear flatter.

Original F=682

F=782

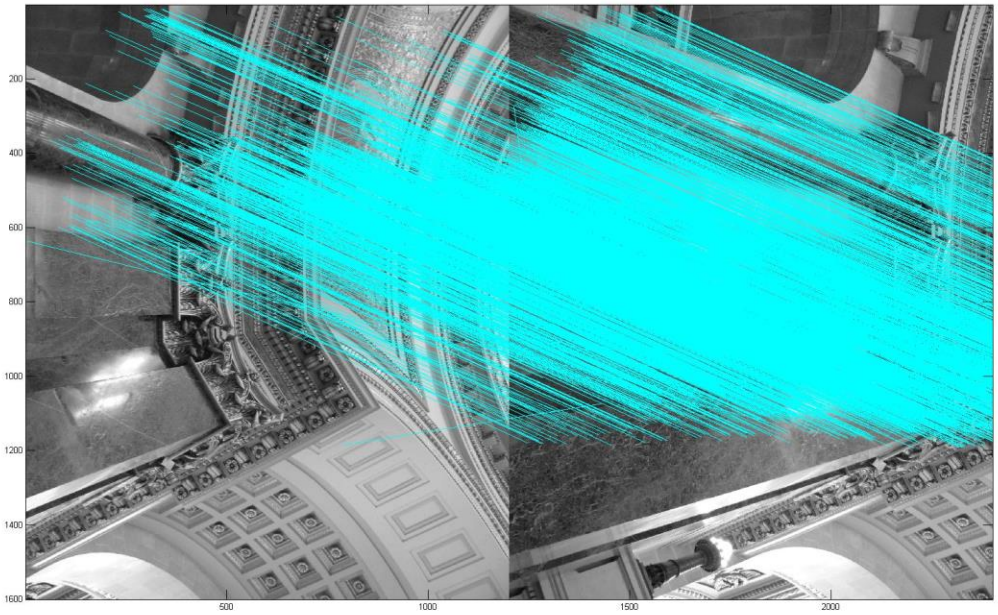


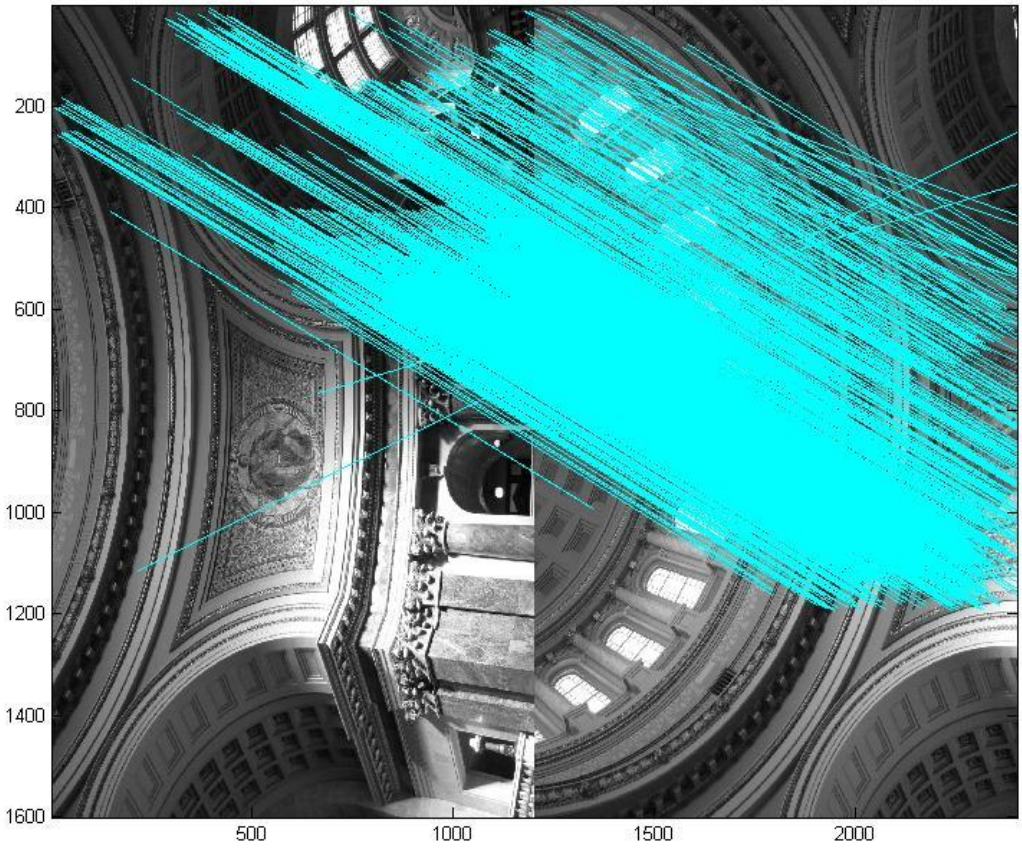
F=582

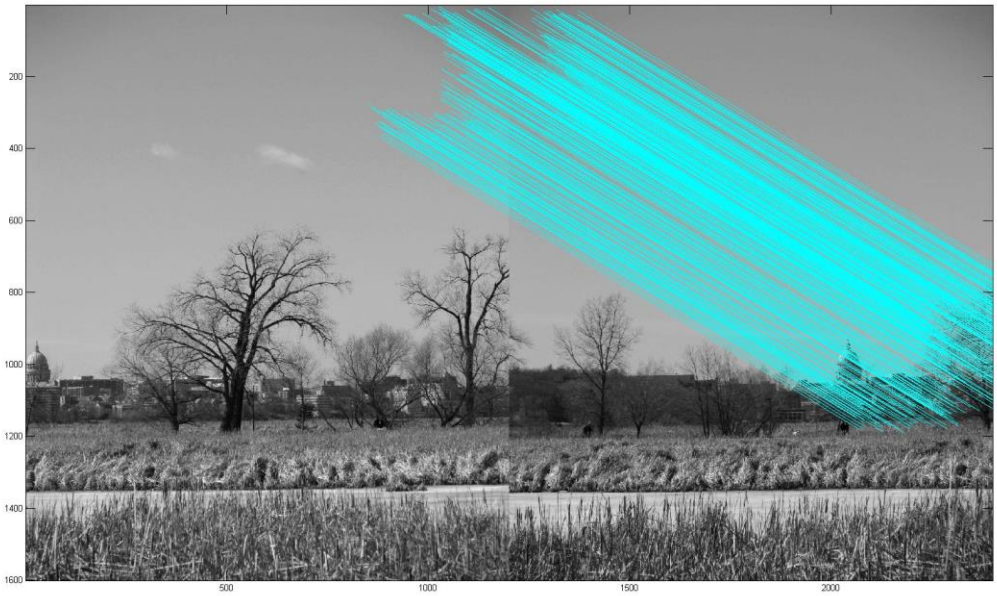
### d. SIFT:-

Detect **SIFT** key points in both images and compute the set of matching points

We use the David Lowe's SIFT Keypoint detector programme which in a very straight forward manner computes the matching points and integrated that into our Matlab setup. It is located under the siftDemoV4 directory under the main directory. The binary Windows executable is called from the mymatch.m function which calls it and it called by the SiftMatch function. Here are some of the outputs of the keypoint detector.




200
400
600
800
1000
1200
1400
1600
500
1000
1500
2000


200
400
600
800
1000
1200
1400
1600
500
1000
1500
2000

### e. RANSAC methods

Ransac was used to estimate homography and then images were projected to have overlap of common points detected in the previous SIFT Keypoint detector programme.

Calculates a transformation that aligns the points points1 and points2 using RANSAC.

We have enabled RANSAC to operate on Affine and Homography transformations based on what is suitable.

**The method returns-**

T: a TFORM object encapsulating the transformation from image1 (points1) onto image2 (points2).

Best Points: Points used to estimate best transformation (Dim: n_pts x 4)

*Some of the intermediate images at this step.*

### f. Stitching & Blending

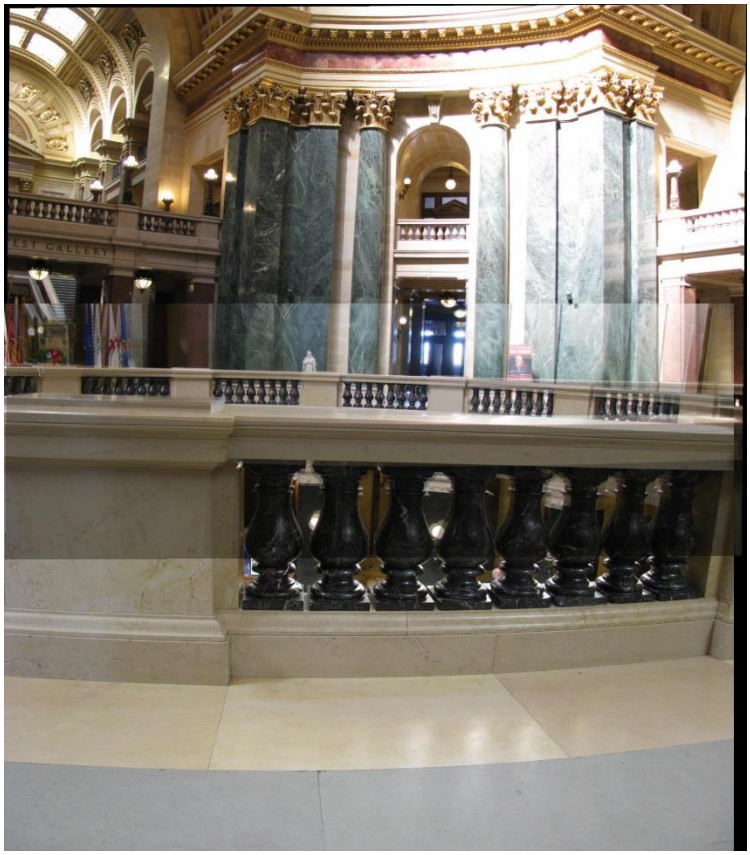
Given two images im1 and im2 and a transformation T_im2 that shall be applied to im2, this function computes a composite image and a mask. The mask 'stitched_mask' is a matrix of the same dimensions as the resulting image whose pixel values are the number of image layers on each pixel.

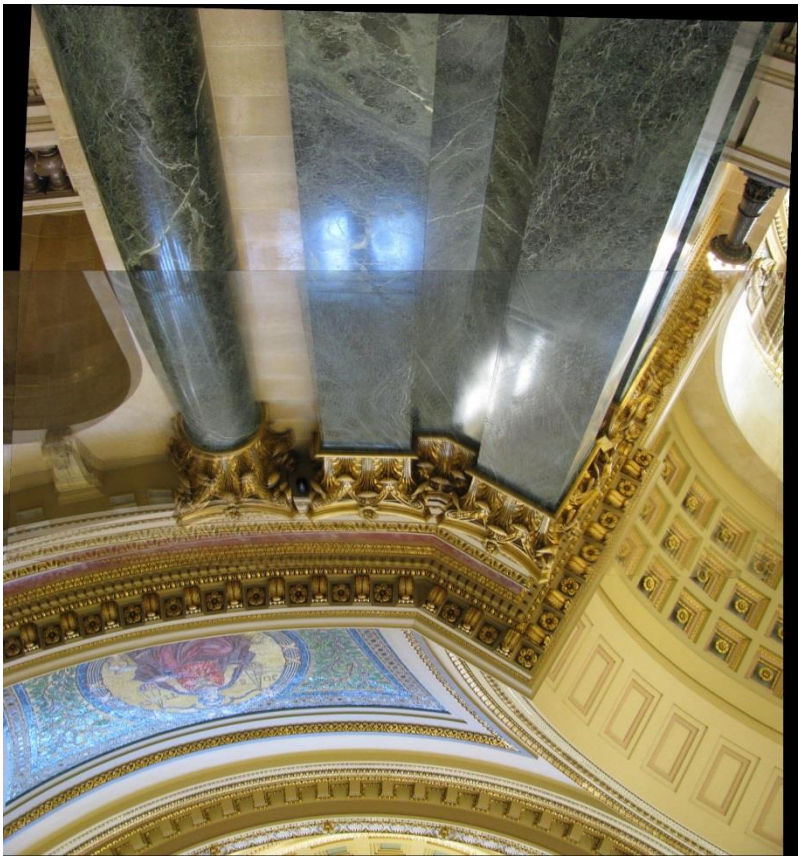
A mask is initialized, and the 2nd image is transformed so that fits on image 1. Then the bounds for the stitched and the input images are found after which the images are aligned with respect to the bounds, to result in an overlapped image which is the stitched version of the 2 images.

Then blending is applied to smoothen out the images. Pyramid and Feather blending have been tried out. The blending files are part of the bells and whistles component and have been put there.

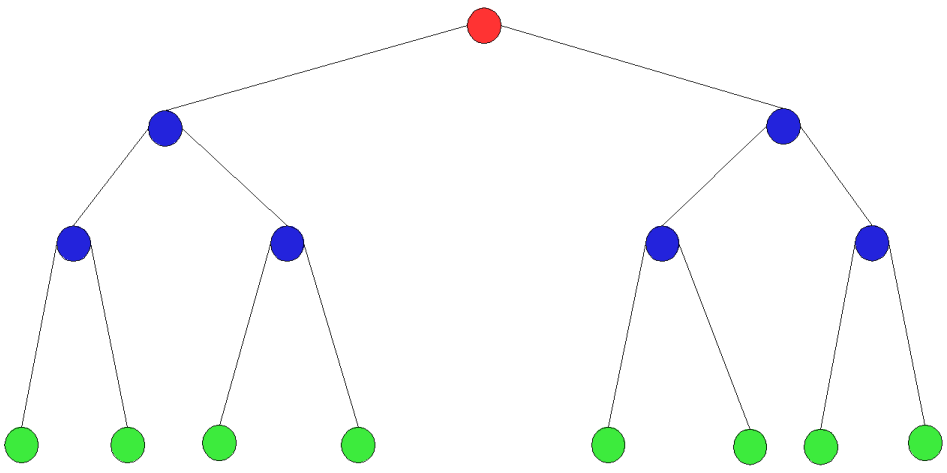
Some of the intermediate outputs are presented below.

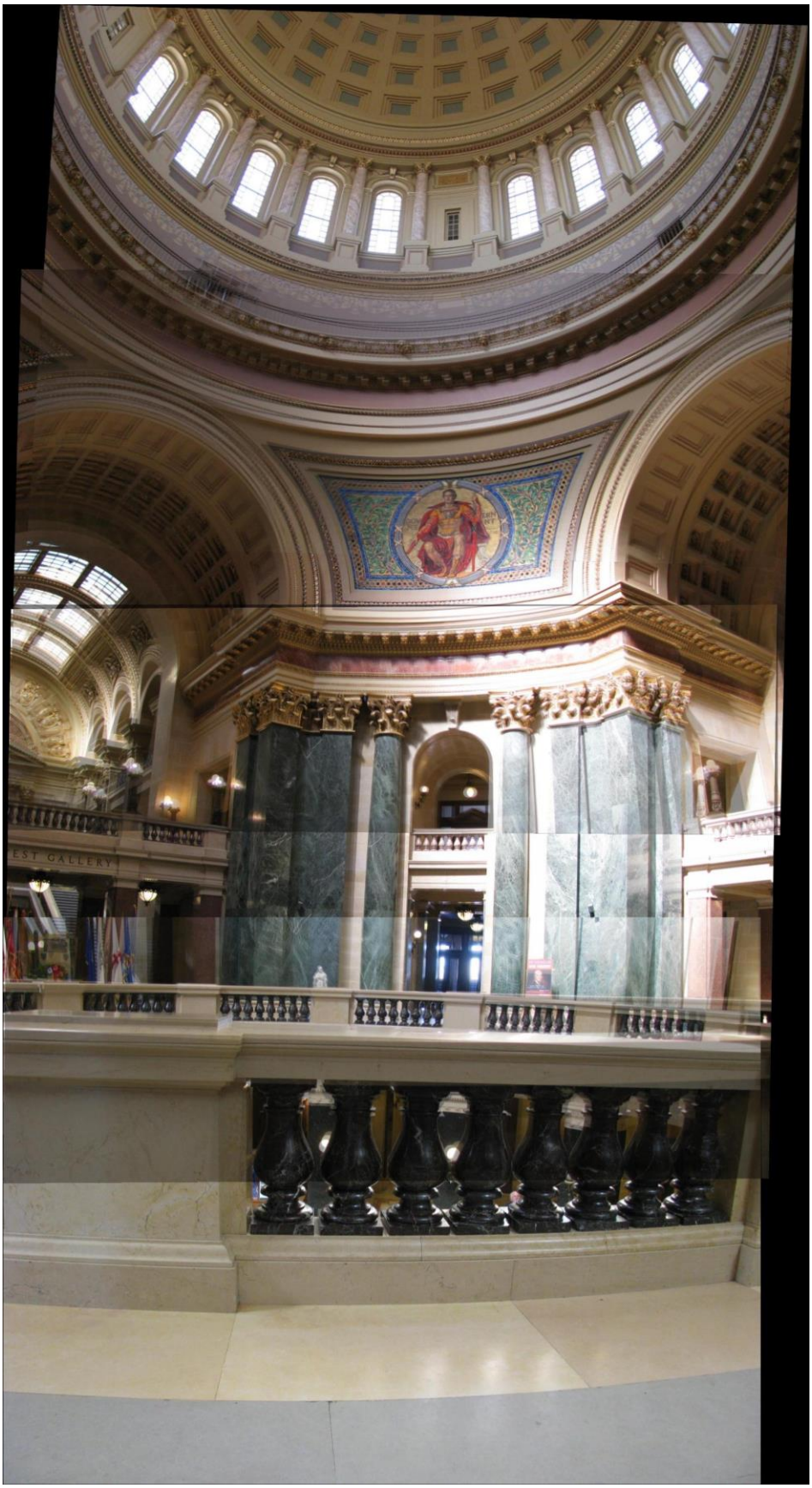
Please note that all these images are before blending and smoothening function was applied.

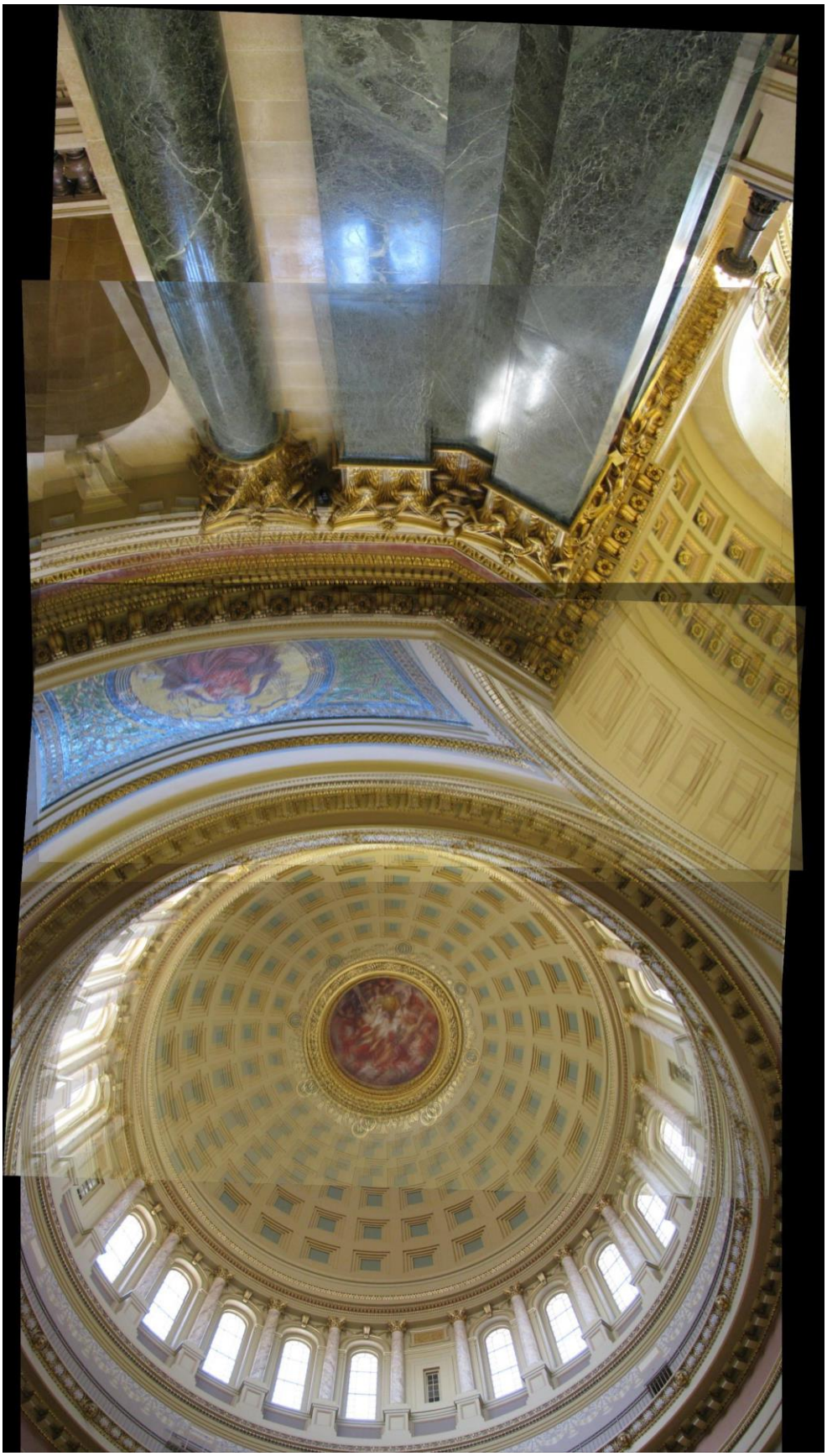




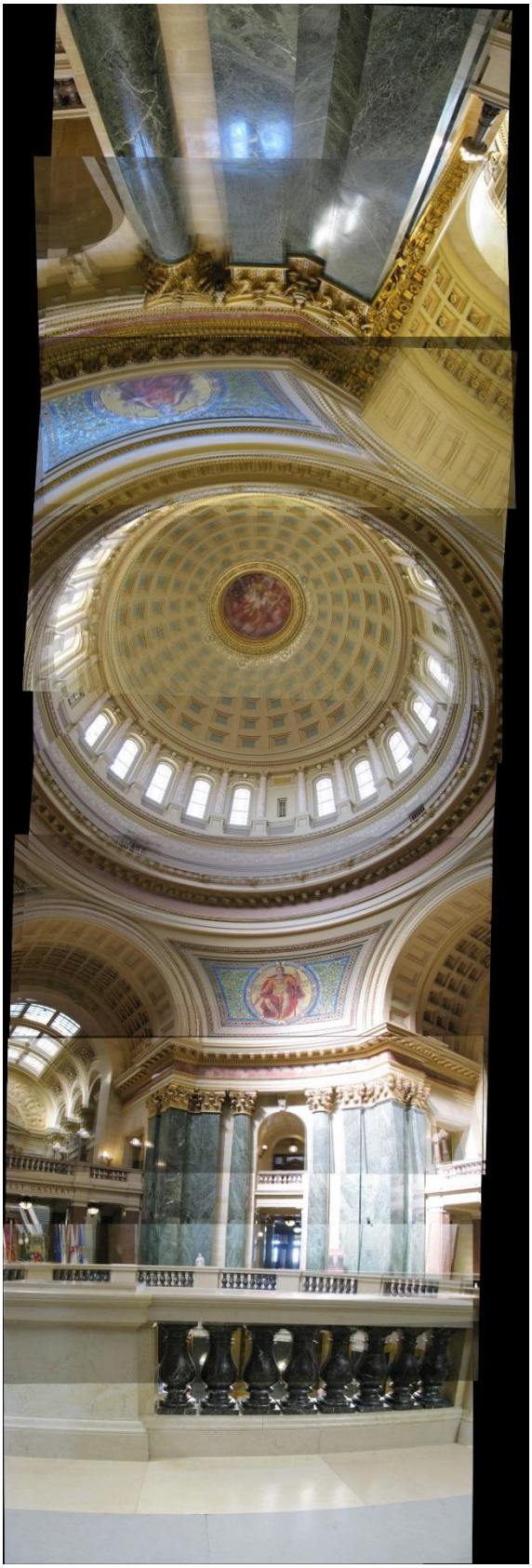
These images were constructed by pairwise stitching input images. We used a binary tree approach to construct the panorama, where we stitch 2 images to form the parent image and keep going upwards till we reach the root image (the full panorama). Unfortunately I was not able to automate the full process and though we could have implemented a linear frame by frame attached to the mother image we felt this methodology would be neater and become computationally more efficient. The process kind of looks like the image below.







Final Constructed Panorama (pre-blending):-



Panorama after blending (no cropping) - please note the last frame was manually readjusted to fit in better:-

Panorama (after cropping and applying some *post-processing touches*, including exposure control):-

Final image, put in horizontal because most of the online viewers wouldn't accept a vertical panorama. This would be the vote image.

**It may be viewed online at:-**

*http://www.dermandar.com/p/bqpMGd?a=0.5*

Select higher resolution version from the options at the bottom of the page.

## 5. Bonus

### a. Pyramid Blending

Pyramid blending is useful in combining two images while keeping the significant features of both the input images. The image is downsized to multiple levels using Gaussian. And then the Gaussian is expanded to the lower lever followed by subtraction from the image on that level. This gives the Laplacian image. Laplacian pyramids for the overlap images are obtained. Then the partial images are combined on multiple Laplacian levels. Then the process of expanding from the top-most to the next level and adding to the original Laplacian image in the same level is performed repeatedly to obtain the final pyramid-blended image.

We have used 6 levels of pyramids. The boundary of separation in the mask needs to be specified by the user.

### b. Tilt-shift Image : Application of Pyramid Blending

Tilt-Shift effect is used to create a miniature scale effect. The resulting image of a life-size real scene appears miniature in its scale. This is achieved by manipulating the focus of the camera. Additionally, the effect is enhanced by increasing saturation and contrast of the image. The effect best materializes in high angle images of buildings, cars, trains and people.

We tried to create the tilt-shift effect in a regular image by implementing pyramid blending. The method involved blending a blurred image with the original image retaining the area of interest and blurring the rest. Further, the saturation values have been enhanced to make it appear miniature-like.

### c. Image compositing

The goal of image compositing is to insert a new image into an existing one and blend the inserted segment so that the final image appears like it is part of the original image. This is an application of image blending.

Per the code written, the user would be able to select region of interest by drawing a polygon over the image to be inserted. Depending upon the region of interest selected, binary masks will be formed for both the input images and the images are blended into a single unit based on the masks.

The code has been written for a specific case particularly which means that for a new set of images, the scale, translation or rotation would be needed to be changed to get proper blending.

## 6. Results

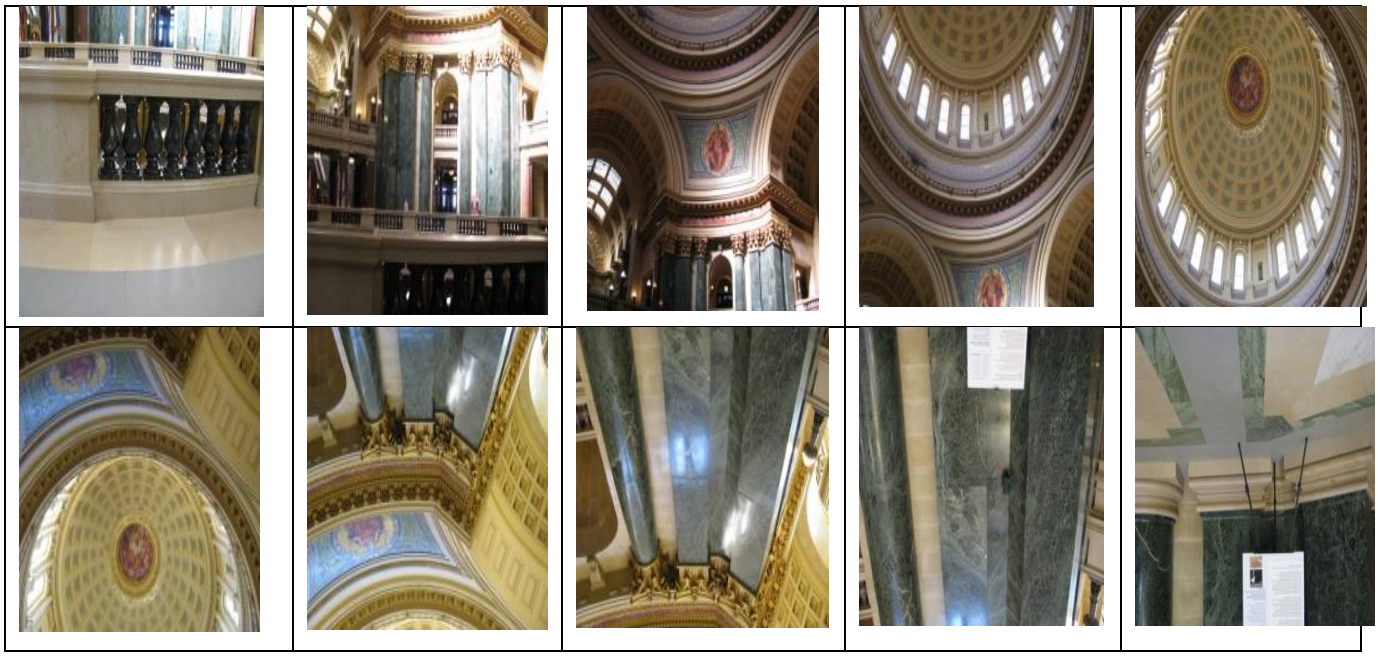
### a. Image Acquisition

#### (i) Madison State Capitol:

We wanted to capture a full view of the ceiling of the state capitol building starting from the west gallery to the east gallery. The focal length, as specified on canvas, was taken to be 682.05 mm for the set of camera that we utilized.

The batteries died midway, thus, there were only limited number of images we could take of the interior of the state capitol.

Following is the input dataset:



#### (ii) Parking Lot overlooking the WIMR building: (Partial view shown)

Here, we wanted to a 360 degree panorama. Covering the state capitol building, Ebling library, WIMR building. In order to capture the images at an appropriate scale, we used

zoom in which invalidates the usage of the given focal length to be incorporated in the cylindrical projection step.

However, since with zooming into the scene, the focal length would increase. Thus, a focal length of **782mm** was used by approximation.

**b. Cylindrical coordinate projection**
Please refer to the section on core Panoramas.

**c. Image pair alignment**
Please refer to the section on core Panoramas.

**d. Stitching and cropping**
Please refer to the section on core Panoramas.

**The final panorama can be viewed in the panoramic view using the following embedded code:**

*<iframe src="//www.dermandar.com/p/bgpMGd?a=0.5" scrolling="no" width="800px" height="449px" style="width:800px; height:449px; overflow:auto; border:0px none transparent; background-color:rgba(0,0,0,1); padding:0px; margin:0px;" frameborder="0"></iframe>*

*Or it may be viewed online at:-*

*http://www.dermandar.com/p/bgpMGd?a=0.5*

**e. Bonus 1: Pyramid Blending**

Input images: taken from Google images:

| Input 1 | Input 2 |
| --- | --- |
|  |  |

Image obtained by copy-pasting the images side-by-side:

Image obtained after pyramid blending with number of pyramids = 6:

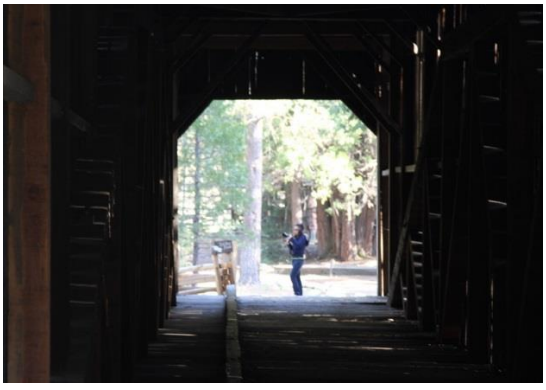
### f. Bonus 2: Tilt-shift Effect: Application of Pyramid Blending

Input Images (taken from Google images): MillenniumPark, Chicago

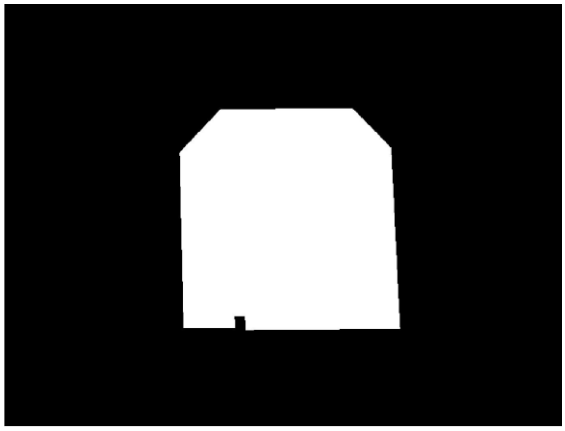
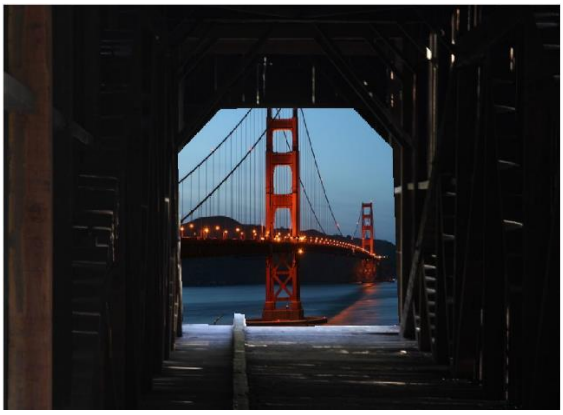
Output image with a basic tilt-shift effect:

**Bonus 3: image compositing**

Input images (taken from example shown on canvas):

| Target Image | Source Image |
| --- | --- |
|  |  |

Region-of-interest is created on target image where the source image would be inserted.

| Region-Of-Interest : Binary Mask | Final Image |
| --- | --- |
|  |  |

Input images (taken from Google images):

| Target Image | Source Image |
| --- | --- |
|  |  |

| Region-of-interest : Binary Mask | Final Image |
| --- | --- |
|  |  |

## 7. Miscellaneous Notes

- Image capturing while pretty simple, in hindsight presented a lot of interesting learning points.
    - We, by the natural placement of the kaidan captured vertical frames in the camera sensor and hence some of our images didn't overlap although we took about 40 snaps in 360 implying that even 10 degree rotation wasn't enough in some scenes to capture meaningful overlaps.
    - We couldn't reconstruct the panorama using that set of images, captured near UHS and had to resort to indoors of Madison capitol for imagery. Turns out it was beautiful too!
    - Also the different in different images was a blessing in disguise. While the raw panorama came out looking not so good looking, we were able to post process on it and convert it to an aesthetically better looking one.
    - This was enabled due to capturing in auto mode which redid the exposures for best lighting condition for the individual images.
- One of the images we took unfortunately got blurred (human hands to blame?). This image was difficult to operate RANSAC upon and we ended up having to discard it. It looks like the features on the blurry image wasn't good enough for the feature matcher to operate on.
- We found later on that we hadn't taken the images with the camera zoomed all the way out (this was necessary due to our relative positioning in the scene). Hence we worked out a mode to bypass the image the cylindrical projection method and instead use homography to project the images onto a common plane and then apply transformation while stitching.
- We used, what we consider to be a neat trick, instead of stitching images in a linearly incremental fashion. We did a binary tree kind of stitching and we think it might be computationally more efficient. We plan to do further analysis to see whether this is indeed true. More details are presented in the core panorama's stitching portion of the report.
- It was interesting to try out various focal lengths for the cylindrical generation function and see the relationship between it and the resulting image and the zoom. Details in the relevant section.

**Individual contributions**

a. Anmol Mohanty: Image acquisition, image panorama with and without cylindrical projections, final write-up, Miscellaneous notes.
b. Rosaleena Mohanty: Image acquisition, pyramid blending, tilt-shift effect, image composition, final write-up

**References**

**a. Panorama:**

R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and texture-mapped models, SIGGRAPH 1997, pp251-258.

M. Brown, D. G. Lowe, Recognizing Panoramas, ICCV 2003.

**b. Pyramid Blending**

P. J. Burt and E. H. Adelson. "The Laplacian pyramid as a compact image code," IEEE Truns. Commun. vol. COM-31, pp. 532-540. Apr. 1983.

**c. Tilt-shift Effect**

en.wikipedia.org/wiki/Tilt–shift_photography