Tonia Elengikal
Sandeep Vanka

May 5, 2020

CS 445 Final Project Submission
Blend Into the Picture

Table of Contents

Motivation

The motivation behind this project is to give users the ability to view old photos of childhood homes/rooms as realistic as possible. There are people who have images of their childhood bedroom, or of their favorite living room way back from before the first digital camera was made. This program allows a user to view 3D rooms in Blender, which is more modern than other free 3D rendering software packages such as VRML. The challenge in this project comes from the use of Blender which we have had the opportunity to work with in this class. We wanted to improve our skills in Blender for a more complicated use case. Additionally, this project will give us experience with projections and applying homographies on images where possible.

Approach

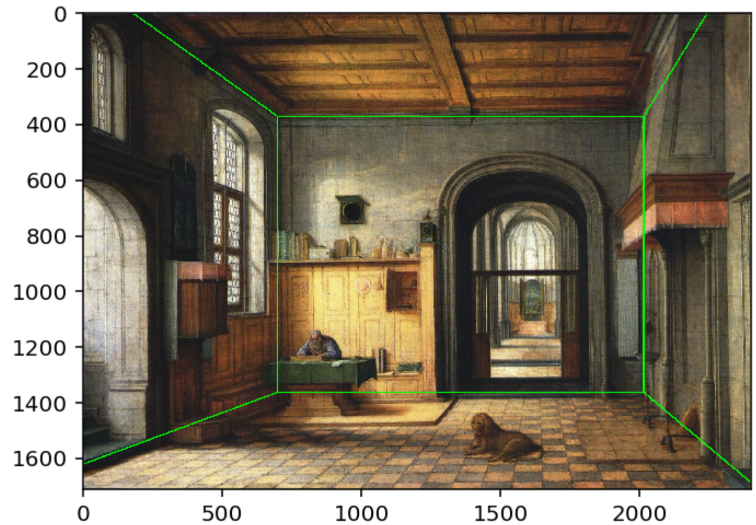
As part of the final project, we have implemented parts of Tour Into the Picture (TIP) algorithm as described in the paper by Horry et al. Using the submitted Jupyter notebook and using the free version of Blender (version 2.82) users can view images in a transformed into a realistic (as possible) 3D room. Our approach of creating 3D scenes utilizes Blender's ability to show planes with applied textures. Our main objective was to take a single view image and transform it into 5 walls to create a 'room' and output obj and mtl files so that Blender can create the planes and applied textures to create 3D scenes. To do this we have done the following over the course of 5 weeks.

Results

First Example – Room

Original Image

Showing Different Planes in Original Image



Homography Images

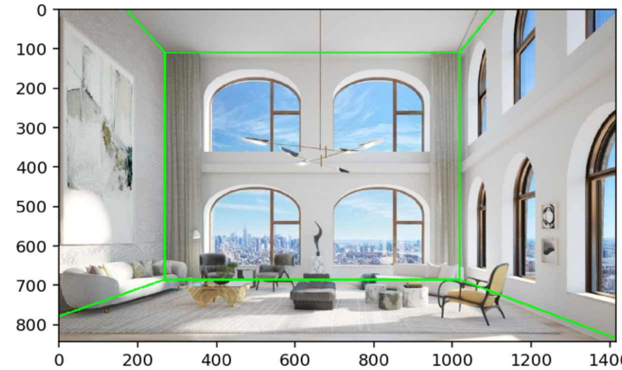
Foreground Object Image

Blender Images

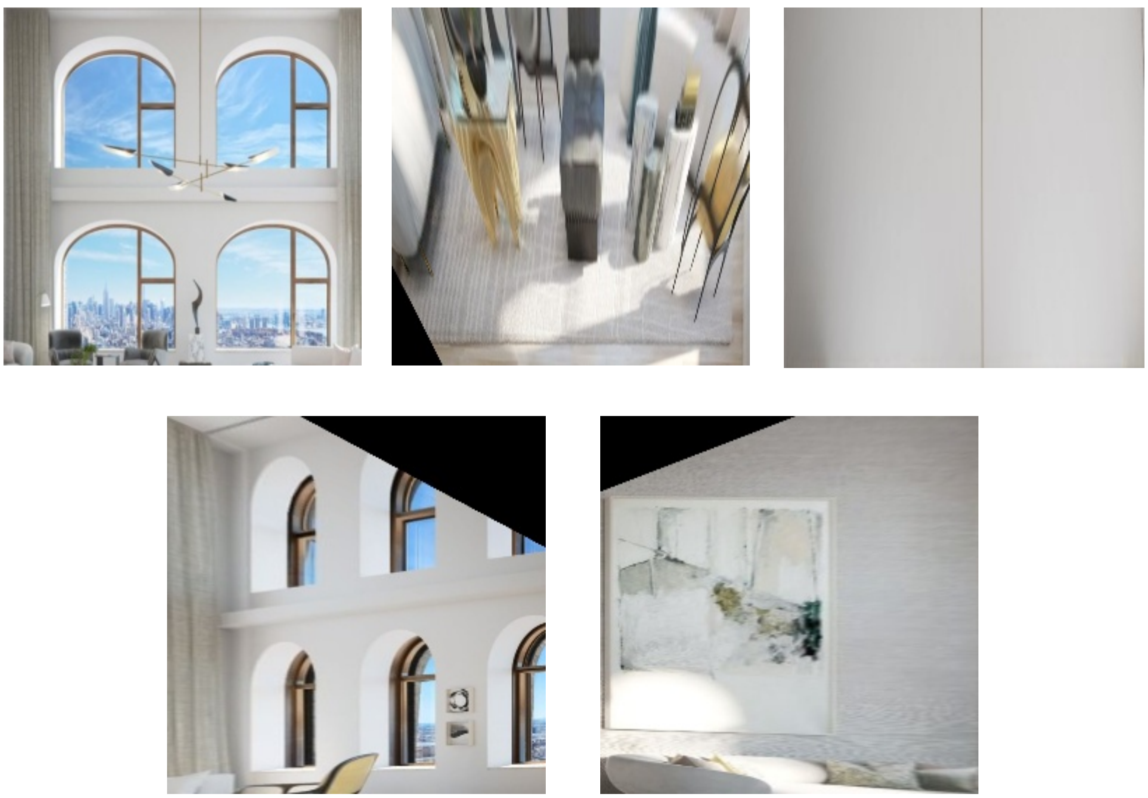
Second Example – Apartment

Original Image

Showing Different Planes in Original Image



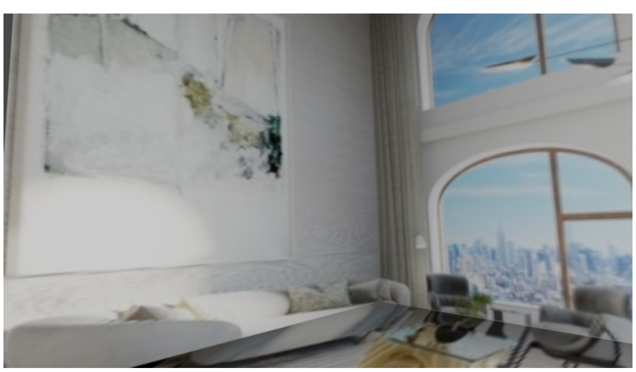
Homography Images

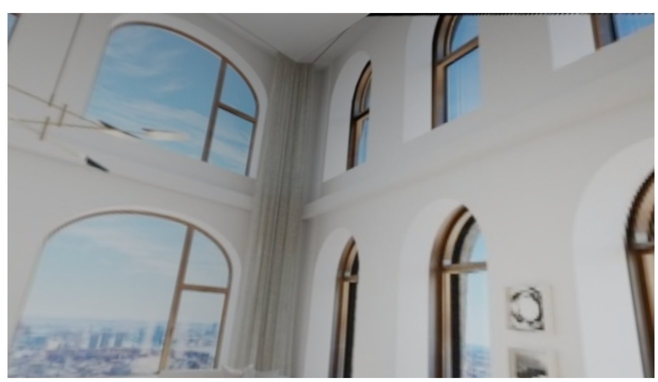


Tonia Elengikal
Sandeep Vanka

May 5, 2020

Blender Images

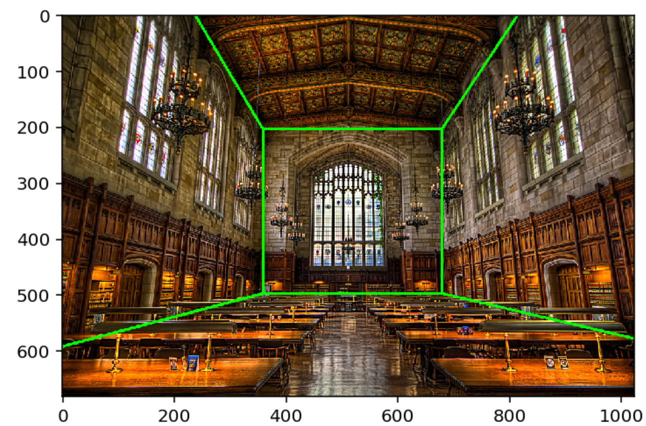




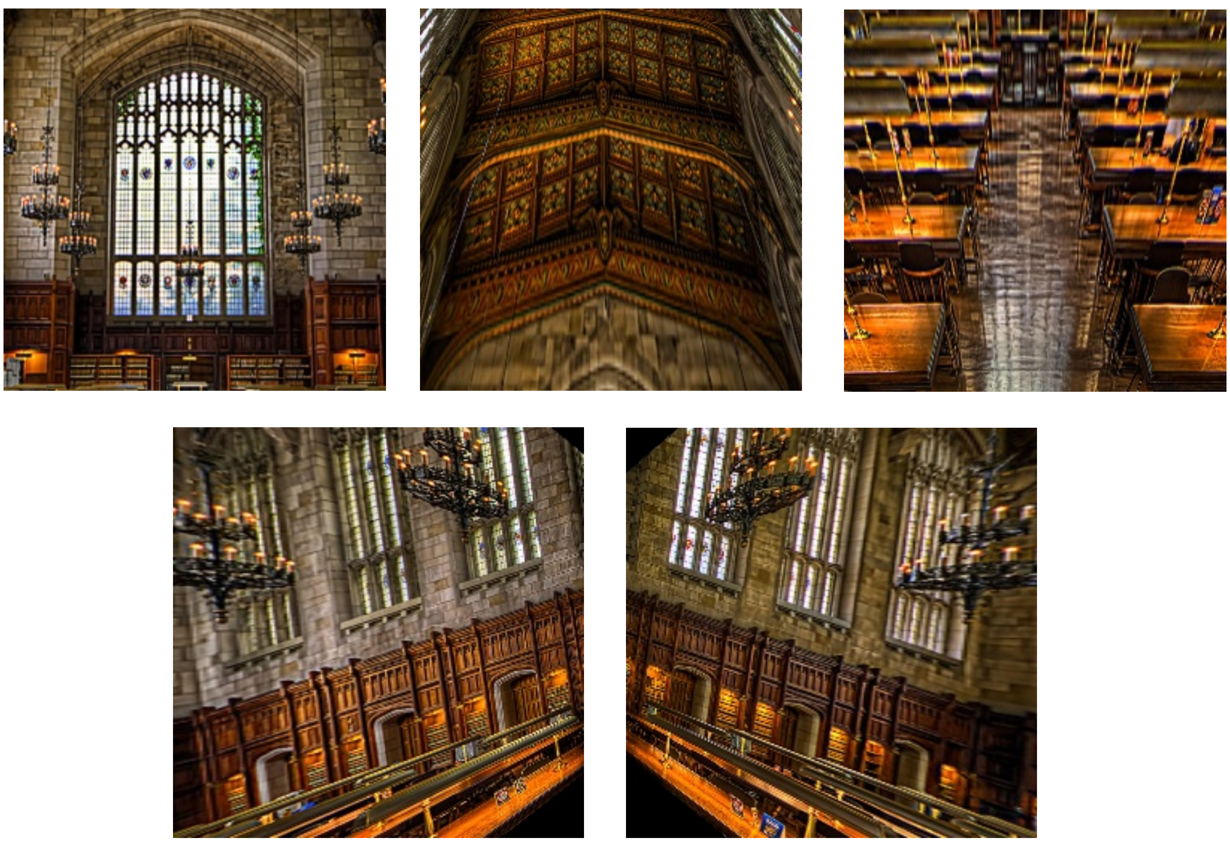
Third Example - Library

Original Image

Showing Different Planes in Original Image



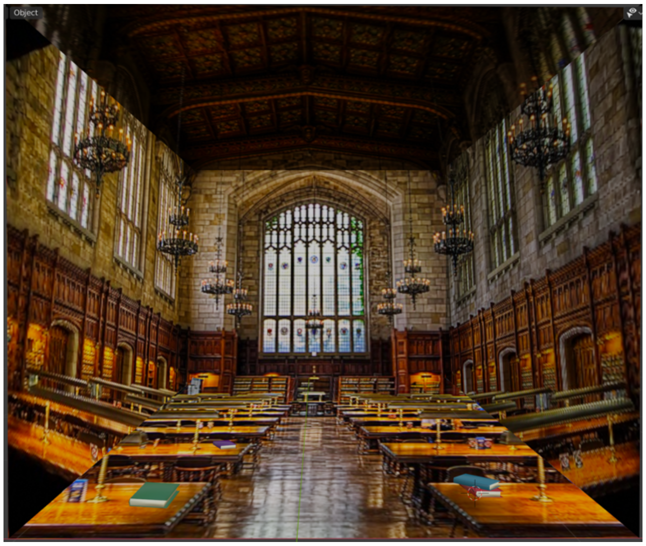
Homography Images

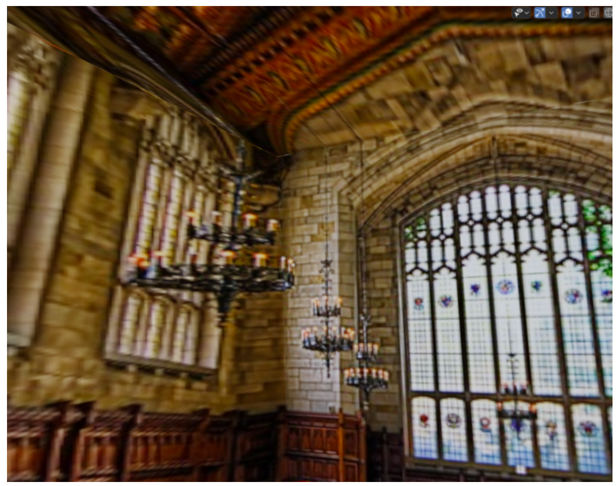


Tonia Elengikal
Sandeep Vanka

May 5, 2020

Blender Images



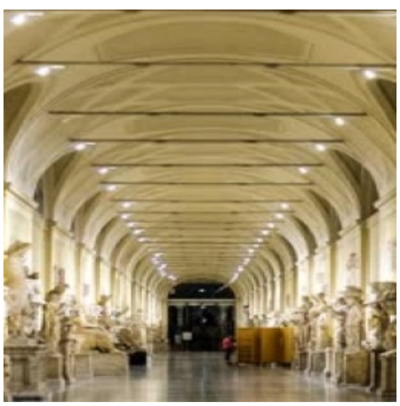


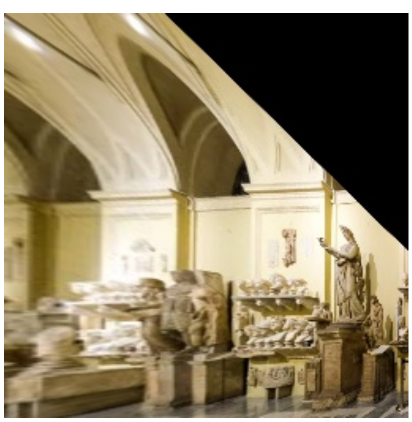
Adding a point source light:

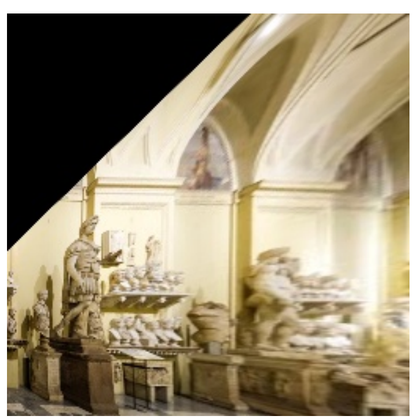
Fourth Example - Museum

Original Image

Homography Images







Blender *Images*

Implementation Details

Implementation was separated into three phases. We considered the first phase to include taking an image and getting the coordinates of the 12 vertices as described in figure 4(c) of the paper.

The first phase included the following steps

1. Have user choose three points, indicating the top left of the room (labeled as point 7 in figure 4 c), vanishing point, and the bottom right of the room (labeled as point 2 in figure 4 c). The vanishing point would need to be roughly estimated by the user based on where the parallel lines in the room are intersecting.
2. Using the three points, the four image vertices are determined as explained in the paper
3. Then using the focal length – provided by the user, but estimated for the images we've used from other sources – the depth is calculated.
4. To calculate the two other points of each of the five planes, 2 lines were created using the vanishing lines and the four corners of the back wall. See figure 4(a) and 4(b) for the lines. From the lines we found where they intersected with the image plane (these are the lines determined by the sides of the image)

The second phase included taking those 12 image points and converted them to 3D world coordinate points

1. The calculated depth in the first phase and setting the camera position to (0,1,0) - and aligning the camera center to vanishing point – automatically determines the four 3D points of the back wall.
2. To determine the rest of the 2 points for each plane, we assumed that the camera plane is parallel to the back wall and it perpendicular to the rest of the four walls. This automatically puts each of the walls on single plane axis (I.e. y=5, x=10).
3. We've also assumed that the camera is looking down –z axis while lying on the z=0 plane. Except for the back wall, the rest of the walls are either on some x or y plane.
4. Once we determined the 3D coordinates of all 12 points, we've created the obj files that contained the four vertices determined each of the faces. The function **saveObjPlane** takes the four planes and the name of the plane and correctly writes the obj file down on disk. This function also writes down the mtl file defining the name of the image that defines the texture file. The obj file references the mtl file to be used for the texture properties. In this case the texture file simply just refers to the image of the wall.
5. The texture files are simple the plane extracts from the raw image with the homography applied. The homography converts that plane into strictly 256x256 resolution images as Blender expects square texture files.
6. For the sake of simplicity we've hard coded the names of the four planes to be, backwall, rightwall, leftwall, topwall, bottomwall. Any name except these will be considered as a foreground object.

The third phase included taking the obj files and importing them into Blender

1. Start a new Blender file and delete the cube object
2. Click File -> Import to import left wall, right wall, back wall, bottom wall obj files into Blender
    a. Make sure to select "Y Forward" and "Z up" as values in transform while importing
    b. The mtl files will automatically be imported with the obj files
3. Add foreground object
    a. Import foreground object Obj file and use the scissor tool to crop it into the shape
    b. If there is no foreground object, add your own by downloading a 3D object .blend file and appending it to your blend file
4. Change camera angle and position by modifying X, Y, Z locations in Blender
5. Move lighting to the middle of the scene and increase to 300 W
6. Click View -> Viewpoint -> Camera to see view from camera and Render -> render image to save image as output file

Challenge / Innovation

Our project was innovative since we used Blender to tour into the picture. One challenge was understanding the OBJ and MTL file formats and creating those files with the correct values in our Python code. Another challenge was the foreground object implementation. It was difficult to determine the correct height of the foreground object we are inserting into the picture. Making the foreground object look realistic, even after cropping it and shaping it, was also a challenge. Another challenge was determining the best focal length – for images that we got from the web or from our personal gallery – to prevent creating planes that are too deep, stretching out the image or having the. Using Blender helped us be innovative since we were able to place 3D objects into the scene and make it look realistic. We were also able to add point source lighting into our scene to make the scene brighter. In the implementations of Tour Into the Picture that we have seen online, no one has used Blender to make the 3D scene. This makes our approach innovative. As a result, we expect to receive 20 points for the challenge/innovation component of grading.

What worked? What didn't work?

Many of the implementations from the paper worked really well with the images we've chosen. The homographies when applied on the planes for the painting, law library, apartment looked realistic at different angles. Adding 3D objects into the 3D rooms worked flawlessly as evidently seen in the apartment and the law library examples. The foreground image for the painting example was okay as Blender expected square textures while planes that we created were in rectangular form. There were instances such as the museum example where the back wall was physically in the far distant, and this made the rest of the walls expand and distort the texture images to a point where we lose resolution. This resulted in low resolution images. We worked around this by creating the back wall close, and setting the backwall in the middle of the hallway so that the walls didn't stretch too far into the back. This resulted in mediocre but acceptable viewpoints.

Future Work

One of the ideas we had was to remove the foreground object on the image and apply hole filling algorithm (which was part of project 3 Bells and Whistles) so that when we created the 3D room it would be without the foreground object. Another idea we had was to incorporate a lightmap into the scene so that lighting was realistic and any 3D objects placed within would be reflected accordingly.

References

http://graphics.cs.cmu.edu/courses/15-463/2006_fall/www/Papers/TIP.pdf

http://paulbourke.net/dataformats/mtl/

Resources

https://www.flickr.com/photos/71336639@N06/7504653198

https://www.cgtrader.com/items/112249/download-page

https://www.cgtrader.com/items/2309336/download-page

https://130william.com/#craft

https://www.indiegamewebsite.com/2020/04/09/the-procession-to-calvary-review/

https://www.wetheitalians.com/default/6-beautiful-italian-museums-visit-free-online